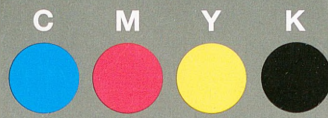




Grey Scale #13



Part Code ST1316



A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19



# AKADEMIA OBRONY NARODOWEJ

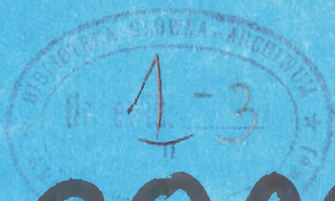
WYDZIAŁ WOJSK LOTNICZYCH I OBRONY POWIETRZNEJ

~~Do użytku służbowego~~

Egz. nr ....

Mjr nawig. dr inż. Piotr MAKOWSKI

OCENA MOŻLIWOŚCI DYŻUROWANIA W POWIETRZU  
LOTNICTWA MYŚLIWSKIEGO  
ZA POMOCĄ PROGRAMOWANIA KOMPUTEROWEGO  
"STREFA"



# 63908

Biblioteka Główna  
Akademii Obrony Narodowej  
S/2834



05-002834-001-0

WARSZAWA

1995



# AKADEMIA OBRONY NARODOWEJ

WYDZIAŁ WOJSK LOTNICZYCH I OBRONY POWIETRZNEJ

~~do użytku służbowego~~

Egz. nr ....

Mjr nawig. dr inż. Piotr MAKOWSKI

OCENA MOŻLIWOŚCI DYŻUROWANIA W POWIETRZU  
LOTNICTWA MYŚLIWSKIEGO  
ZA POMOCĄ PROGRAMOWANIA KOMPUTEROWEGO  
"STREFA"



63908

Biblioteka Główna  
Akademii Obrony Narodowej  
S/2834



05-002834-001-0

WARSZAWA

1995

WYDZIAŁ WOJSK LOTNICZYCH I OBRONY POWIETRZNEJ



~~.....~~

Egz. nr .....

OCENA MOŻLIWOŚCI DYŻUROWANIA W POWIETRZU  
LOTNICTWA MYŚLIWSKIEGO  
ZA POMOCĄ PROGRAMOWANIA KOMPUTEROWEGO  
"STREFA"

Autor:

mjr nawig. dr inż. Piotr MAKOWSKI

Konsultacja naukowa:

prof. dr hab. inż. Stefan ANTCZAK

Konsultacja informatyczna:

mjr dr inż. Andrzej GRZELKA



## SPIS TREŚCI

|  | str. |
|--|------|
| WSTĘP .....  | 4    |
| 1. PODSTAWOWE WSKAŹNIKI OKREŚLAJĄCE MOŻLIWOŚCI DYŻUROWANIA<br>W POWIETRZU I SPOSOBY ICH OKREŚLANIA ..... | 6    |
| 1.1. Ocena stosowanych metodyk określania możliwości<br>dyżurowania w powietrzu .....                    | 6    |
| 1.2. Propozycje wykorzystania technik symulacyjnych ....   | 14   |
| 2. ANALIZA CZYNNIKÓW WPLYWAJĄCYCH NA MOŻLIWOŚCI<br>DYŻUROWANIA W POWIETRZU SAMOLOTÓW MYŚLIWSKICH .....   | 17   |
| 3. MODEL MATEMATYCZNY PROCESU DYŻUROWANIA W POWIETRZU ....   | 24   |
| 3.1. Założenia przyjęte podczas tworzenia modelu .....   | 24   |
| 3.2. Dane potrzebne do wykonania obliczeń .....  | 26   |
| 3.3. Charakterystyka startu i lotu wznoszącego .....   | 28   |
| 3.4. Charakterystyka lotu poziomego z ustaloną<br>prędkością .....                                       | 29   |
| 3.5. Charakterystyka zniżania i zająścia do lądowania ...  | 30   |
| 3.6. Charakterystyka manewru w płaszczyźnie poziomej ...   | 31   |
| 3.7. Bilans paliwa - pozostałość paliwa dysponowana na<br>dyżurowanie .....                              | 31   |
| 3.8. Obliczanie możliwości dyżurowania jednostek LM ....   | 32   |
| 4. OGÓLNY ALGORYTM REALIZACJI OBLICZEŃ MOŻLIWOŚCI<br>DYŻUROWANIA W POWIETRZU .....                       | 37   |
| 5. OPIS UŻYTKOWANIA MODUŁU PROGRAMOWEGO -"STREFA"<br>I PRZYKŁAD OBLICZEŃ .....                           | 40   |
| 5.1. Podstawowe wiadomości o programie .....   | 40   |
| 5.2. Przykład obliczeń .....   | 43   |

BIBLIOGRAFIA .....44

ZAŁĄCZNIKI

1. WYBRANE WYDRUKI TREŚCI ZOBRAZOWANYCH NA EKRANIE  
MONITORA PODCZAS UŻYTKOWANIA PROGRAMU "STREFA" .....45

2. WERSJA ŹRÓDŁOWA PROGRAMU "STREFA" - WYLISTOWANE  
PROCEDURY PROGRAMOWE (ROZSZERZENIE "PAS") .....66

## WSTĘP

Jak wykazują doświadczenia licznych ćwiczeń dowódczo-sztabowych i gier wojennych realizowanych w ramach szkolenia w WLOP i AON jednym z zasadniczych sposobów działań lotnictwa myśliwskiego (LM) w systemie obrony powietrznej RP było przechwytywanie środków napadu powietrznego z dyżurowania w powietrzu. Wynikało to z ograniczonych możliwości pozyskania uprzedzającej informacji o przeciwniku powietrznym na dalekich podejściach do nakazanych rubieży wprowadzenia do walki LM. Położenie geopolityczne Polski i oparcie systemu rozpoznania sytuacji powietrznej o naziemne środki radiolokacyjne, tworzące strefę informacji radiolokacyjnej, zwykle nie wystarczająca do radiolokacyjnego zabezpieczenia działań LM z dyżurowania na lotniskach, zmusza do stosowania mniej ekonomicznych sposobów działań, polegających na przechwytywaniu przeciwnika z dyżurowania w powietrzu. Planując działania bojowe LM bardzo ważną zatem okazuje się znajomość jego możliwości dyżurowania w powietrzu (w wyznaczonych strefach dyżurowania).

Obliczenia tych możliwości wykonuje się na wszystkich szczeblach dowodzenia lotnictwem podczas planowania działań bojowych w celu wstępnego określenia możliwości czasowych niezbędnych do analizowania wariantów prowadzenia walki i oceny oczekiwanych jej rezultatów.

Realizacja tych obliczeń polega na sporządzeniu modelu lotu bojowego zgodnego z decyzją dowódcy i rozliczenia zużycia paliwa na poszczególnych jego etapach zgodnie z obowiązującymi metodykami prowadzenia kalkulacji inżynierjno-nawigacyjnych.

Przedsięwzięcie to jest jednym z zasadniczych zadań zabezpieczenia nawigatorskiego działań bojowych tego lotnictwa w okresie planowania działań.

Duże prawdopodobieństwo popełnienia błędów przypadkowych i znaczna pracochłonność podczas wykonywania kalkulacji dotyczących określania czasów dyżurowania w powietrzu, powinny być pokonywane poprzez tworzenie stosownego oprogramowania komputerowego.

Pomyślne rozwiązanie tego problemu w formie wprowadzenia do eksploatacji stosownych procedur programowych, których użytkowanie byłoby możliwe na wszystkich szczeblach dowodzenia lotnictwem niewątpliwie ułatwiłoby proces planowania działań (zapewniłoby jednoznaczność oceny własnych możliwości przestrzenno-czasowych na każdym szczeblu dowodzenia).

Niniejsze opracowanie jest kolejną próbą wyjścia na przeciw oczekiwaniom służby nawigatorskiej w zakresie wspomagania komputerowego realizacji zadań zabezpieczenia nawigatorskiego działań bojowych LM, którego rezultatem jest przykład praktycznego rozwiązania omawianych zagadnień w formie procedury programowej pk. "STREFA".

# 1. PODSTAWOWE WSKAŹNIKI OKREŚLAJĄCE MOŻLIWOŚCI DYŻUROWANIA W POWIETRZU I SPOSOBY ICH OKREŚLANIA

## 1.1. Ocena stosowanych metodyk określania możliwości dyżurowania w powietrzu

Możliwości dyżurowania LM w powietrzu charakteryzują zdolność omawianego lotnictwa do realizacji zadań utrzymywania określonych sił w wyznaczonych strefach dyżurowania w konkretnej sytuacji operacyjno-taktycznej. W teorii zdolności te wyraża się za pomocą wskaźników będących umownymi miarami liczbowymi.

Podstawowe wskaźniki określające możliwości LM dyżurowania w powietrzu dają odpowiedź na najczęściej spotykane pytania-problemy w procesie podejmowania decyzji, podczas oceny sytuacji.

Pytania te można sformułować następująco:

1. Jak długo można utrzymywać określone siły w wyznaczonej strefie dyżurowania wyznaczając do tego zadania pewną liczbę samolotów (wraz z załogami)?

2. Jaka liczbę samolotów potrzeba zaangażować do zadania utrzymywania w danej strefie dyżurowania grupy o określonym składzie, w nakazanym okresie czasu (lub ciągle)?

3. Jak długo może być utrzymywana w wyznaczonej strefie pojedynczy samolot (para)?

Pierwszy i trzeci wskaźnik jest wyrażany w jednostkach czasu, drugi natomiast liczbą samolotów.

Czas dyżurowania samolotu w strefie dyżurowania, jest to maksymalny czas jaki może przebywać samolot w strefie dyżurowania, w zależności od odległości strefy od lotnisk z

uwzględnieniem zużycia paliwa na walkę powietrzną i na powrót na własne lotnisko przy nakazanej pozostałości paliwa.

Należy podkreślić, że znajomość czasu dyżurowania pojedynczych samolotów (grup samolotów) w wyznaczonych strefach jest wskaźnikiem niezbędnym do określania wartości pozostałych wskaźników.

Określanie czasów dyżurowania grup samolotów w wyznaczonych strefach jest problemem, który w praktyce rozwiązuje się poprzez korzystanie ze sporządzanych dla danych typów samolotów określonych wykresów. Przykłady ich znajdują się między innymi na s.34, s.63, s.49, s.50 pozycji "METODYKA OBLICZEŃ INŻYNIERYJNO-NAWIGACYJNYCH ORAZ ZESTAW WYKRESÓW DLA SAMOLOTÓW BOJOWYCH", Lot 1928/79, oraz przedstawia jeden z nich **rysunek 1**.

Wykresy te sporządzono metodami kalkulacyjnymi między innymi przy założeniach, że:

- lot do strefy i powrót odbywa się na tej samej wysokości co wysokość dyżurowania w strefie;
- prędkość dolotu do strefy, dyżurowania w niej i powrotu jest tak zwaną prędkością maksymalnego zasięgu w warunkach atmosfery standardowej.
- jednostkowe zużycie paliwa podczas dyżurowania w strefie równe jest zużyciu paliwa podczas lotu prostoliniowego w warunkach maksymalnego zasięgu.

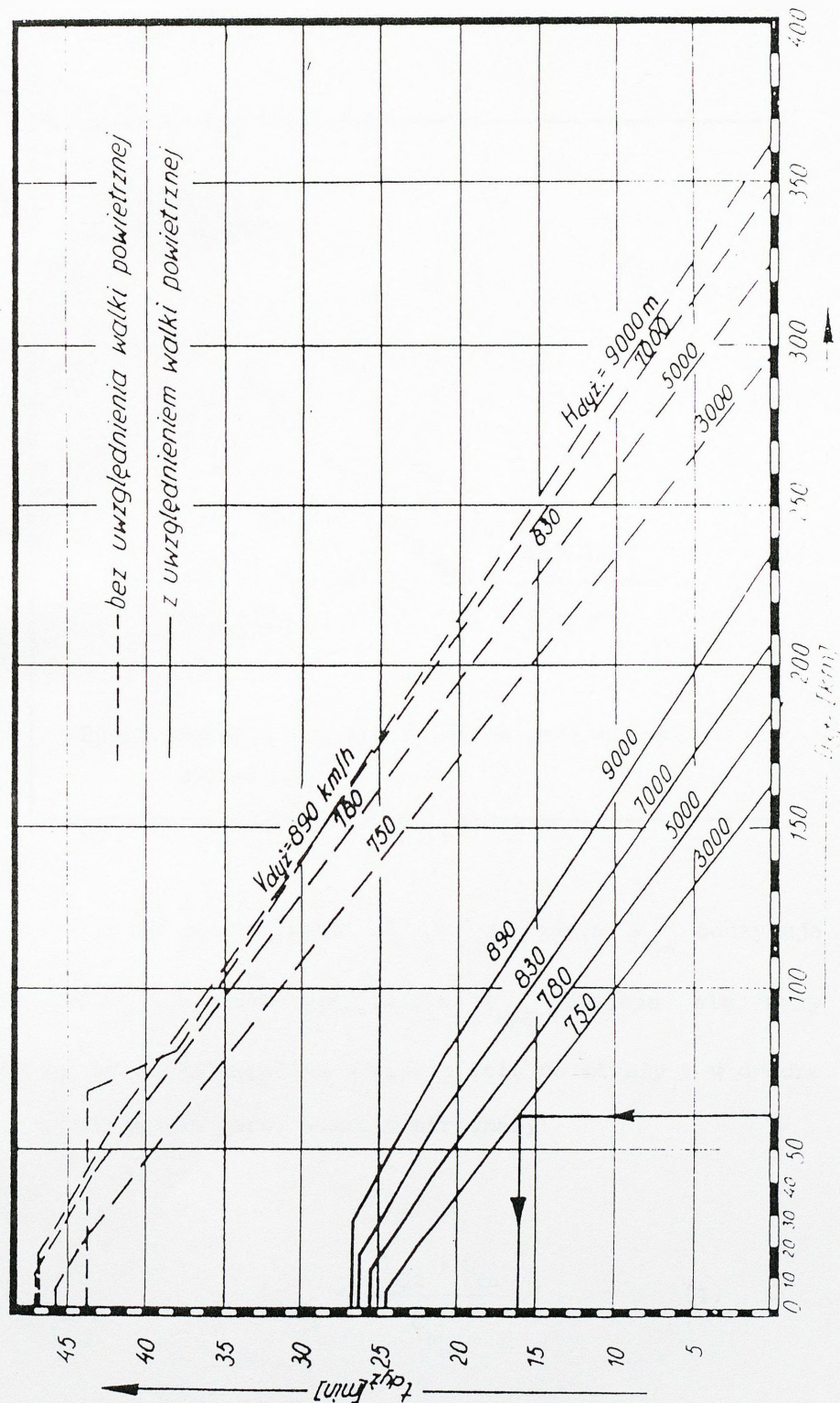
Obliczenia do wspomnianego wykresu, zgodnie z definicją wykonuje się w następujący sposób:

- na osi odciętych odклада się odległość do strefy oraz taktyczny promień działania;

ZŁS. WYKRES CZASÓW DYŻUROWANIA SAMOLOTU MiG-21M (MiG-21MF)  
ZE ZBIORNIKIEM DODATKOWYM O POJEMNOŚCI 765 l. Z DWOMA R-3S (RS-2US) I Z DWIEMA UB-16-57U

Całkowity zapas paliwa 3365 l. Ciężar właściwy paliwa 0,775 g/cm<sup>3</sup>. Nawigacyjny zapas paliwa 260 l. Rezerwowy zapas paliwa 500 l. Start i wzniesienie na pełnym dopalaniu. Wykonanie walki powietrznej w ciągu 3 min na pełnym dopalaniu i 3 min pracy silnika. Wykonanie kręgu do lądowania w ciągu 4 min na wysokości 500 m

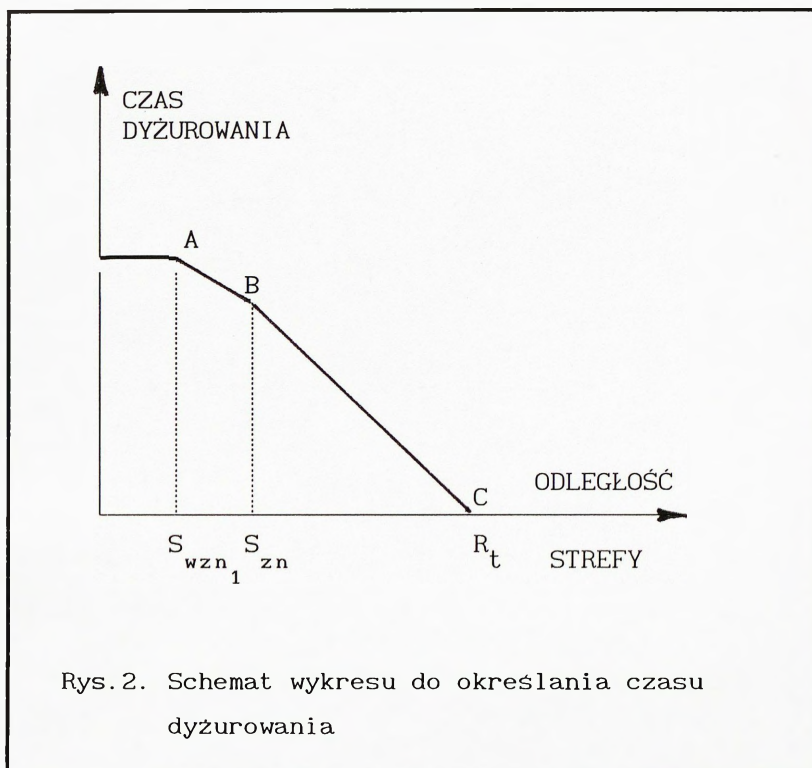
na zakresie maksymalnego zasięgu lotu. Prędkość dyżurowania, lot do strefy i powrót na zakresie maksymalnego zasięgu lotu. Zauważenie na minimalnym zakresie pracy silnika. Wykonanie kręgu do lądowania w ciągu 4 min na wysokości 500 m



Rys.1. Zbiór wykresów do określania czasów dyżurowania dla samolotu MiG-21.

- na osi rzędnej odklada się czas dyżurowania w strefie.

Dla warunku, gdy droga wznoszenia jest mniejsza od drogi zniżania, współrzędne punktów A,B,C (rys.2.) oblicza się w następujący sposób:



- punkt A ma współrzędne  $(S_{wzn_1}, t_{dyż_1})$ , gdzie  $S_{wzn}$  odczytuje się z tabel, a czas dyżurowania  $t_{dyż_1}$  oblicza się przy założeniu, że zużycie paliwa w czasie lotu do strefy i w drodze powrotnej równa się zero. Wówczas otrzymamy:

$$t_{dyż_1} = \frac{Q_{dyż} + Q_{zn}}{C_{hmz}} \quad (1)$$

gdzie:

$Q_{dyż} = Q_{1p}$  - zużycie paliwa w czasie dyżurowania oblicza się ze wzoru:

$$Q_{1p} = Q_1 - Q_z - Q_{poz} - Q_{wzn} - Q_{zad} - Q_{zn} - Q_{kr};$$

$C_{hmz}$  - godzinowe zużycie paliwa na zakresie maksymalnego zasięgu lotu;

- punkt B ma współrzędne  $(S_{zn}, t_{dyż}_2)$ , gdzie  $S_{zn}$  odczytuje się z tabel, a czas dyżurowania  $t_{dyż}_2$  oblicza się przy założeniu, że zużycie paliwa w czasie lotu do strefy i w drodze powrotnej równa się zero. Wówczas otrzymamy:

$$t_{dyż}_2 = \frac{Q_{dyż}}{C_{hmz}} - \frac{(S_{zn} - S_{wzn}) \cdot C_{kmz}}{C_{hmz}} \quad (2)$$

gdzie:

$C_{kmz}$  - kilometrowe zużycie paliwa na zakresie maksymalnego zasięgu lotu;

- punkt C ma współrzędne  $(S_{max} = R_r, t_{dyż}_3 = 0)$ , gdzie uzyskuje się maksymalną odległość strefy działania od lotniska równą taktycznemu promieniowi działania, przy założeniu, że czas dyżurowania, zużycie paliwa w czasie lotu do strefy i w drodze powrotnej równa się zero. Wówczas otrzymamy:

$$S_{max} R_r = \frac{Q_{dyż}}{2C_{kmz}} + \frac{S_{wzn} + S_{zn}}{2} \quad (3)$$

Dla warunku, gdy droga wznoszenia jest większa od drogi zniżania, nie oblicza się punktu A, lecz punkty B i C.

Współrzędne punktów B i C oblicza się w następujący sposób:

- punkt B ma współrzędne  $(S_{wzn}, t_{dyż})$ , przy poprzednich założeniach i oblicza się ze wzoru:

$$t_{dyż_2} = \frac{Q_{dyż}}{C_{h_{mz}}} + \frac{(S_{wzn} - S_{zn}) \cdot C_{k_{mz}}}{C_{h_{mz}}} \quad (4)$$

- punkt C ma współrzędne  $(S_{max} = R_1, t_{dyż_3} = 0)$  przy poprzednich założeniach i oblicza się ze wzoru 2.

Biorąc pod uwagę, że czasy dyżurowania w strefach zależą nie tylko od warunków lotu (wysokości, prędkości lotu) ale także od rodzaju manewrów, ładunku bojowego, związanego z nim paliwa przewidywanego na walkę powietrzną, gęstości paliwa, warunków atmosferycznych oraz warunków dolotu i powrotu ze stref. Wykonanie pełnej gamy omawianych charakterystyk uwzględniających wymienione uwarunkowania, z uwagi na ich "astronomiczną liczbę", wydaje się niecelowe, wobec oczywistych trudności z ich praktycznym wykorzystywaniem. Chociaż względy merytoryczne jednoznacznie przesadzają o potrzebie uwzględniania tych wszystkich uwarunkowań.

Głównym zarzutem kierowanym pod adresem wykorzystywanych metod graficznych jest fakt nie uwzględniania w kalkulacjach, że podczas wykonywania manewrów (kursem) w strefie dyżurowania, jednostkowe zużycie paliwa znacząco wzrasta (w wyniku utraty części siły nośnej). Potwierdzają to analizy charakterystyk

lotno-taktycznych samolotów. W zależności od warunków lotu wzrost ten może być rzędu i 100%. Ponieważ średni łączny czas manewrów statów powietrznych wykonujących zadania dyżurowania w powietrzu wynosi około 30-40% ogólnego czasu przebywania w strefie nie uwzględnianie opisywanych zjawisk w kalkulacjach musi prowadzić do powstania błędów.

Inną alternatywą w stosunku do metod graficznych jest metoda analityczna polegająca na sporządzeniu obliczeń inżyniersko-nawigacyjnych lotu zgodnie z obowiązującymi instrukcjami. Metoda ta jest jednak niezwykle czasochłonna co bardzo ogranicza jej praktyczne stosowanie.

Czas dyżurowania w powietrzu grupy samolotów ( $t_d$ ) jest również potrzebny do określania możliwości ciągłego dyżurowania ( $T_d$ ) pododdziałów lub oddziałów LM. Literatura przedmiotu podaje następującą zależność, pozwalającą szacować ten czas:

$$T_d = \frac{N_m}{n_{gr}} \cdot t_d \cdot K ; \quad (5)$$

gdzie:  $N_m$ -ogólna liczba samolotów (planowanych wylotów);  
 $n_{gr}$ -liczba samolotów jednocześnie dyżurujących w strefie;  
 $K$ -współczynnik zmienności dyżurowania.

Zależność (5) nie uwzględnia **problematyki odtwarzania gotowości bojowej** lądujących kolejno samolotów, powracających na lotnisko po wyznaczonym czasie dyżurowania. Użytkownik korzystając z niej zakłada, że odtworzenie gotowości bojowej powracających samolotów

będzie niemożliwe przed terminem ich ponownego startu. Biorąc pod uwagę, że "grafik dyżurowania" będzie w warunkach bojowych realizowany zgodnie z planem dopóki nie pojawi się przeciwnik powietrzny, należy sądzić, że powracające ze stref grupy nie wykorzystają uzbrojenia. Ich czas odtwarzania gotowości bojowej będzie zatem stosunkowo krótki, zdeterminowany głównie ponownym tankowaniem paliwa. Stąd w wielu wypadkach odtworzenie gotowości bojowej będzie możliwe, a wartość rezultatów otrzymanych w wyniku korzystania z przytoczonej zależności mocno zaniżona. W tej sytuacji wydaje się potrzebnym opracowanie metody, uwzględniającej model odtwarzania gotowości bojowej samolotów myśliwskich. Analogiczne potrzeby stwierdzono przy realizacji zadania pochodnego, polegającego na określaniu potrzebnej liczby sił do utrzymywania określonych grup samolotów w strefach dyżurowania (Nm).

Analiza matematycznych podstaw stosowanych metodyk podczas określania wyżej opisanych wskaźników wykazuje, że stosowanie metod kalkulacyjnych wydaje się niekorzystne w przypadku etapu oceniania czasu dyżurowania pojedynczego samolotu (grupy samolotów). Wynika to z konieczności stosowania pewnych założeń upraszczających. Dla przykładu, na każdym odcinku lotu poziomego trzeba określić średnie jednostkowe zużycie paliwa, które jak wiadomo zmienia się wraz ze zmianą masy statku powietrznego spowodowanej wypracowywaniem paliwa. Jeżeli zatem nie wiadomo jak zmieni się masa samolotu nie można dokładnie szacować jednostkowego zużycia paliwa i odwrotnie. Analogiczne problemy występują podczas matematycznych prób opisu innych etapów lotu, takich jak start, wznoszenie, zniżanie, manewr kursem czy

ładowanie. Powstają więc pytania:

- które czynniki mają znaczący wpływ na możliwości dyżurowania w powietrzu LM ?

- jakie metody określania wskaźników możliwości dyżurowania w powietrzu LM mogą stanowić alternatywę w stosunku do stosowanych metod graficzno-kalkulacyjnych ?

Kolejny rozdział 3 niniejszego opracowania jest próbą odpowiedzi na pytanie pierwsze.

Jeśli zaś chodzi o odpowiedź na pytanie drugie to zespół autorski powziął przypuszczenie, że podczas określania wskaźników możliwości dyżurowania w powietrzu LM metody symulacyjne mogą okazać się korzystniejsze od metod stosowanych dotychczas.

#### 1.2. Propozycje wykorzystania technik symulacyjnych

Spośród wielu możliwości wykorzystania metod i technik symulacyjnych w rozwiązywaniu omawianych problemów jedną z prostrzych metod wydaje się wykorzystanie komputerowych modeli samolotów myśliwskich (np.: MiG-21 i MiG-29). W celu określania czasu dyżurowania pojedynczych samolotów (grup) modele te mogłyby w czasie nierzeczywistym realizować wielokrotnie: start, lot po zadanej trasie do strefy dyżurowania, określone manewry w strefie oraz powrót na zadane lotnisko i lądowanie. W pierwszym "wylocie" czas dyżurowania byłby bliski zera, w kolejnych zwiększałby się o pewną wartość  $\Delta T$  proporcjonalną do pozostałości paliwa ( $Q_p$ ) jaka algorytm zapamiętał po ostatnim "wylądowaniu". Zwiększanie czasu dyżurowania trwałoby oczywiście do momentu gdy wartość pozostałości paliwa po kolejnym "lądowaniu" będzie bliska

wartości ustalonej  $Q_{ust}$  (np. paliwo potrzebne na dodatkowy manewr lądowania) z tolerancją  $\pm b$ . Wówczas ostatni czas dyżurowania odpowiadałby poszukiwanemu wskaźnikowi. Dokładność tej metody zależy oczywiście od adekwatności modeli samolotów w stosunku do samolotów rzeczywistych a także od wartości liczbowych parametrów  $a, b$  zależności (6) i (7).

$$\Delta T = a \cdot Q_p; \quad (6) \quad \text{abs}(Q_{ust} - Q_p) < b; \quad (7)$$

Im mniejsze będą wartości parametrów  $a, b$  tym dokładność metody będzie większa. W ogólnym przypadku zwiększy się jednak czas obliczeń. Stąd istnieje konieczność empirycznego wyboru tych parametrów w celu osiągnięcia kompromisu między dokładnością a czasem symulacji. Należy podkreślić, że analogiczną metodę do proponowanej autor zastosował do obliczania promienia taktycznego samolotu Su-22M4, w której kolejnemu wydłużaniu podlegała trasa lotu.

Inną możliwością wykorzystania komputerowych modeli samolotów, jaka zarysowała się w procesie analizy matematycznego modelu realizacji zadania dyżurowania w powietrzu, było zastosowanie ich do obliczeń cząstkowych wartości zużycia paliwa na poszczególnych etapach lotu. W wyniku tego powstała metoda niejako kombinowana, łącząca elementy symulacji komputerowej dla potrzeb metody kalkulacyjnej. Porównanie wyników uzyskiwanych tymi metodami (przy wykorzystaniu tych samych komputerowych modeli samolotów) wykazało pomijalne różnice wartości czasów dyżurowania, przy nieznacznie dłuższym czasie kalkulowania (rzędu 2-3 sekund dla IBM DX486) w metodzie zaproponowanej jako pierwsza. Przesadziło to o przyjęciu metody "kombinowanej" do opracowania procedury programowej "STREFA",

której podstawy matematyczne i algorytm a także zasady tworzenia komputerowych modeli samolotów LM przedstawiono w kolejnych rozdziałach.

## 2. ANALIZA CZYNNIKÓW WPŁYWAJĄCYCH NA MOŻLIWOŚCI DYŻUROWANIA W POWIETRZU SAMOLOTÓW MYŚLIWSKICH

Przedmiotem rozważań zamieszczonych w niniejszym opracowaniu są możliwości czasowe wykonania zadania dyżurowania w powietrzu w wyznaczonej strefie przez grupę samolotów MiG-21Bis lub MiG-29.

Zasadnicze czynniki determinujące te możliwości to:

- 1) typ samolotu wyznaczonego do wykonania zadania;
- 2) skład grupy potrzebnej do wykonania zadania;
- 3) wymagany w danej sytuacji operacyjno-taktycznej profil trasy dolotu do nakazanej strefy dyżurowania i trasy powrotnej oraz warunki lotu po trasie, w tym także sposób formowania i rozformowania ugrupowania bojowego, sposób lądowania i wykonania ataku;
- 4) odległość nakazanej strefy dyżurowania od lotniska bazowania (lotnisk startu i lądowania).
- 5) wariant napełnienia instalacji paliwowej (w tym gęstość paliwa) ;
- 6) **bieżące zużycie paliwa podczas lotu na poszczególnych jego etapach.**

Ad.1) Wpływ typu samolotu myśliwskiego na możliwości dyżurowania jest oczywisty. Zależność ta generuje potrzebę posługiwania się właściwymi dla danego typu statku powietrznego metodykami (instrukcjami) obliczania zasięgu i długotrwałości lotu.

Ad. 2) Lot w ugrupowaniu wpływa na zwiększenie rozchodu paliwa. Przyjmuje się, że lot w szyku powoduje zwiększone zużycie paliwa o około:

3% dla klucza;

7% dla eskadry.<sup>1/</sup>

Można zatem wnioskować, że możliwości przestrzenne wykonania zadań bojowych podczas lotu w ugrupowaniu będą maleć w tych samych proporcjach jak wzrasta zużycie paliwa w stosunku do lotu pojedynczego statku powietrznego. Analizy rozwiązań proponowanych w licznych ćwiczeniach dowódczo-sztabowych i grach wojennych wykazały, że podczas dyżurowania w strefach lotnictwo myśliwskie nie stosuje grup wykonujących lot w ugrupowaniach zwartych lub ześrodkowanych większych niż klucz samolotów.

Ad. 3) Przebieg profilu lotu (- liczba manewrów prędkością, etapów wznoszenia i zniżania) ma decydujący wpływ na możliwości przestrzenne wykonania zadania bojowego. Biorąc pod uwagę, że strefy dyżurowania LM podczas wykonywania zadań w systemie OP będą znajdowały się na kierunkach spodziewanego nalotu ŚNP nie ma powodów, dla których wysokość dolotu do strefy dyżurowania i wysokość powrotu powinny być mniejsze od wysokości dyżurowania w strefie. Natomiast dolot i powrót mogą się odbywać na wysokościach większych niż wysokość dyżurowania ze względu na oszczędności w zużyciu paliwa, jeżeli tylko względy taktyczne na to zezwalają, zwłaszcza gdy strefy położone są w znacznych odległościach od lotnisk bazowania.

Formowanie ugrupowań bojowych w składzie nie większym niż klucz, przy typowym sposobie startu - parami odbywa się zwykle metodą dopędzania na trasie lotu do strefy. Wystarczy zatem przyjąć do obliczeń warunki lotu pary dopędzającej. Podobnie

-----

<sup>1/</sup>"Metodyka obliczeń inżyniersko-nawigacyjnych lotu oraz zestaw wykresów dla samolotów bojowych" s.82; Lot. 1928/79. Poznań 1980.

rozpuszczenie grup do lądowania odbywa się zazwyczaj na trasie powrotnej, tak by możliwe było lądowanie z rubieży co 20 sekund. W kalkulacjach możliwości dyżurowania, należy przyjmować warunki lotu, ostatniego z grupy, statku powietrznego.

Zakładany sposób wykonania ataku - prowadzenia walki powietrznej jest ściśle związany z ilością paliwa rezerwowaną na te cele. Masa paliwa przeznaczona na prowadzenie walki stanowi zazwyczaj poważną pozycję w ogólnym bilansie paliwa samolotów myśliwskich. Stare metodyki<sup>1/</sup> zalecają obliczanie masy paliwa na walkę powietrzną zakładając 3 minuty lotu na pełnym dopalaniu i 3 minuty lotu z prędkością maksymalnego zasięgu. W tym przypadku, na małych wysokościach masa paliwa walki powietrznej samolotu MiG-21Bis osiągnęłaby wartość około 1000kg, a samolotu MiG-29 mocno by ją przekroczyła. Stanowi to okło 35% masy paliwa zbiorników zasadniczych. Liczne opracowania teoretyczne dotyczące prowadzenia walk powietrznych nie dają jednoznacznej odpowiedzi jak kalkulować masę omawianego paliwa. Najbardziej predystynowana osobą do składania propozycji w tym zakresie jest szef strzelania plm, który na podstawie analizy zadania i oceny przeciwnika powietrznego, mając do dyspozycji analizy obiektywnej kontroli lotów szkolnych i bojowych jest w stanie ustalić racjonalną wartość tego zapasu paliwa.

Ad.4) Odległość nakazanej strefy dyżurowania od lotniska bazowania (lotnisk startu i lądowania) jest jednym z ważniejszych czynników determinujących możliwości dyżurowania samolotów w powietrzu. Paliwo dysponowane na dyżurowanie jest bowiem paliwem

-----

<sup>1/</sup>"Metodyka obliczeń inżyniersko-nawigacyjnych lotu oraz zestaw wykresów dla samolotów bojowych" s.82; Lot. 1928/79. Poznań 1980.

pozostałym po sporządzeniu bilansu paliwa potrzebnego na start do lotu do strefy, powrót i lądowanie, walkę powietrzną. W pewnych szczególnych warunkach, gdy wyznaczy się położenie strefy dyżurowania w odległości równej promieniowi taktycznemu samolotów czas dyżurowania w takiej strefie jest równy zeru. Oznacza to, że samoloty natychmiast powinny wejść do walki i po jej stoczeniu powrócić na lotnisko bazowania.

Ad. 5) Wariant napełnienia instalacji paliwowej (możliwość podwieszenia zbiorników dodatkowych paliwa, zrzuconych lub nie zrzuconych po wypracowaniu paliwa) wpływa na możliwości przestrzenne wykonania zadania bojowego. Na przykład podwieszenie na samolocie MiG-21 trzech zbiorników dodatkowych (2x490l i 1x760l) powoduje wzrost zapasu paliwa o 54%. Powoduje to znaczne zwiększenie możliwości dyżurowania. Dla przykładu MiG-21 z dwoma rakietami R-3S, bez zbiorników dodatkowych może dyżurować w strefie oddalonej o 50km (na wysokości 1000m przy założonym paliwie walki powietrznej - 500kg przy stosowaniu ekonomicznych reżymów lotu) przez około 24 minuty. W wariantcie napełnienia ze zbiornikami dodatkowymi (2x490l i 1x760l) czas ten wzrasta do 38 minut. Gęstość paliwa decyduje o jego wartości energetycznej. Spotyka się gęstości paliwa mieszczące się w zakresie od 0.705 do 0.855. Przy napełnieniu instalacji paliwowej paliwami o tak różnej gęstości możliwości dyżurowania mogą w pewnych warunkach zmienić się o około 15% i więcej.

Ad. 6) Bieżące zużycie paliwa na poszczególnych etapach lotu jest obok czynników wymienionych wyżej, czynnikiem decydującym o możliwościach dyżurowania LM. Zależy ono od:  
-warunków lotu (wznoszenie, zniżanie, manewr, lot poziomy

- z określoną prędkością i wysokością lotu);
- temperatury otaczającego powietrza;
- bieżącej masy samolotu podczas lotu;
- właściwości aerodynamicznych podwieszeń zewnętrznych;
- prędkości i kierunku wiatru względem bieżącego kursu samolotu.

Powyższe czynniki są uwzględniane w algorytmach sporządzania obliczeń inżyniersko-nawigacyjnych opisywanych w instrukcjach obliczania zasięgu i długotrwałości lotu. Świadczy to o potrzebie ich uwzględniania podczas kalkulacji przy użyciu programów komputerowych.

**Prędkość lotu** ma znaczący wpływ na możliwości przestrzenno-czasowe LM. Dla przykładu różnica prędkości o 10% w stosunku do prędkości maksymalnego zasięgu podczas lotu MiG-29 na wysokości 200m (bez podwieszeń) powoduje wzrost zużycia paliwa o 10%. Dla większych odchyleń wzrost ten jest znacznie większy ze względu na silnie nieliniowy charakter zależności opisujących zjawisko. Analogiczne właściwości wykazuje w tym zakresie samolot MiG-21.

**Wysokość lotu** znacznie wpływa na możliwości dyżurowania, Im większa wysokość lotu tym możliwości te są większe. Możliwości LM wyposażonego w samoloty MiG-21 i MiG-29 podczas wykonywania lotów bojowych na średnich i dużych wysokościach mogą szacunkowo wzrosnąć o około 100% i więcej w stosunku do możliwości podczas wykonywania lotów na małych wysokościach.

**Temperatura otaczającego powietrza** podczas lotu na stałych wysokościach przy stałych prędkościach (przryzadowych lub Macha) nie wpływa praktycznie na zużycie paliwa (a więc i na możliwości dyżurowania). Wpływ temperatury należy uwzględnić podczas

wznoszenia szczególnie do wysokości średnich i dużych. Wzrost temperatury o około 20 stopni C ponad temperaturę standardową może spowodować wzrost około 20% zużycia paliwa podczas wznoszenia do omawianych wysokości. Ponieważ loty do stref dyżurowania i powrotne, a także sam akt dyżurowania w strefach, są wykonywane z zasady na prędkościach poddźwiękowych ( bez włączania zakresu pracy silnika: "dopalenie") wpływ temperatury na zużycie paliwa podczas rozpędzania jak i sam proces rozpędzania w pierwszym przybliżeniu można pominąć.

Bieżąca masa samolotu ma duży wpływ na zasięg i długotrwałość lotu (w tym na możliwości dyżurowania w powietrzu). Na przykład, podczas lotu poziomego z prędkością maksymalnego zasięgu wpływ masy ( $m_1$  i  $m_2$ ) na kilometrowe zużycie paliwa ( $ck_1$  i  $ck_2$ ) można w przybliżeniu określić w oparciu o zależność (8):

( dla wysokości barycznych mniejszych niż 1000m)

$$ck_2 = ck_1 * [1 + 0.1(m_2/m_1 - 1)]; \quad (8)$$

Podczas lotu z prędkościami odbiegającymi od prędkości maksymalnego zasięgu wpływ masy na zużycie paliwa jest silnie nieliniowy i jak wykazuje analiza stosownych wykresów literatury przedmiotu [3][4] jest większy. Zmiana masy samolotu o 2000kg na wysokości  $H=500m$  powoduje zmianę  $ck$  o 2%, a na  $H=8000m$  o około 16%.

Właściwości aerodynamiczne podwieszonych odwzorowany jest w kalkulacjach inżyniersko-nawigacyjnych lotu tzw. współczynnikiem oporów czołowych (WOC). Współczynniki te przyjmują ściśle określone wartości charakterystyczne dla danego typu uzbrojenia i miejsca podwieszenia (skrzydło, kadłub). Podlegają one z zasady algebraicznemu sumowaniu. Wartości ich i sposób określania



sumarycznego WOC dla samolotu w danym wariantcie uzbrojenia zawarte są w pozycji [3][4]. Podwieszenia zewnętrzne w większym stopniu powodują zwiększenie jednostkowego zużycia paliwa w locie poziomym niż podczas wznoszenia. Ich wpływ na możliwości przestrzenno-czasowe samolotów LM wynikający z oporów aerodynamicznych jest większy niż wynikający ze wzrostu masy samolotu. W warunkach realizacji zadania dyżurowania w powietrzu zrzućenie zbiorników po wypracowaniu, bez kontaktu z przeciwnikiem powietrznym, byłoby oczywistą rozrzutnością.

Prędkość i kierunek wiatru uwzględnia się poprzez wprowadzenie pojęcia wiatru ekwiwalentnego (zminna-  $U_e$ ). Wiatr może poprawiać lub powiększać możliwości przestrzenne samolotów. Kilometrowe zużycie paliwa z uwzględnieniem wpływu wiatru  $ck_w$ , jeżeli trasa lotu składa się z jednakowych odcinków wykonywanych raz z wiatrem a raz pod wiatr, oblicza się z zależności(9):

$$ck_w = ck / [1 - (U_e/V)^2]; \quad (9)$$

gdzie:  $V$  - prędkość powietrzna samolotu [m/s];

$U_e$  - prędkość wiatru ekwiwalentnego [m/s], (może mieć wartość ujemną);

Jak wynika z zależności (7) przy prędkościach poddźwiękowych lotu rzędu 200m/s i prędkościach wiatru rzędu 10m/s zużycie paliwa może zmienić się o około 2.5%. Ponieważ szczególnie podczas manewrowania w strefie wpływ wiatru jest trudny do oszacowania i ze względu na jego niewielki wpływ w typowych warunkach atmosferycznych w pierwszym przybliżeniu można go naszym zdaniem pominąć.

### 3. MODEL MATEMATYCZNY PROCESU DYŻUROWANIA W POWIETRZU

#### 3.1. Założenia przyjęte podczas tworzenia modelu

Podczas wykonywania tych zadań samoloty LM wykonują loty w przedziale małych i średnich wysokości, rzadko na wysokościach dużych, rzędu 9000m. Stąd przyjęto w proponowanym modelu zakres wysokości modelowania ograniczony od dołu wysokością 200m, a od góry 9000m.

Prędkościowy zakres lotu ograniczono od dołu prędkością maksymalnego zasięgu, a od góry  $V=0.90Ma$ . Założono że podczas lotu poziomego zakres pracy silnika "dopalenie" nie będzie stosowany. Wynika to z faktu, że każde włączenie dopalania w sposób znaczący ogranicza możliwości przestrzenno-czasowe samolotów odrzutowych. Na małych wysokościach zużycie kilometrowe paliwa może wzrosnąć około 6 razy, a więc w kilka minut można zużyć cały zapas paliwa dysponowany na lot.

Zakres uwzględnianych zmian temperatury od  $0^{\circ}C$  do  $30^{\circ}C$ .

Biorąc pod uwagę wnioski z analiz typowych planów lotów na wykonanie zadania dyżurowania w powietrzu sporządzanych podczas ćwiczeń taktyczno-bojowych stwierdzono, że model matematyczny typowego lotu na wykonanie zadania bojowego dla samolotów LM dla celów obliczeń możliwości dyżurowania powinien składać się z modeli takich etapów jak:

- start parami;
- wznoszenie do nakazanej wysokości lotu;
- lot poziomy (według warunków ekonomicznych-maksymalnego zasięgu lub z nakazaną prędkością)-dolot do nakazanej strefy

- dyżurowania lub punktu ewentualnego zniżania;
- zniżanie do nakazanej wysokości dyżurowania;
  - dyżurowanie w strefie - 2 minuty lotu poziomego z nakazaną prędkością, zakręt ustalony o kąt 180 stopni z kątem przechylenia 30 stopni, 2 minuty lotu poziomego i kolejny manewr o kąt 180 stopni itd.;
  - wznoszenie do nakazanej wysokości powrotu;
  - lot poziomy (według warunków ekonomicznych-maksymalnego zasięgu lub z nakazaną prędkością) - dolot do początkowego punktu zniżania;
  - lądowanie z rubieży pojedynczo co 20 sekund;
  - odtwarzanie gotowości bojowej;
  - zajęcie gotowości bojowej, (praca silnika samolotów na ziemi);
  - start parami;
  - ...itd.

Na marginesie trzeba zaznaczyć, że rozpuszczenie ugrupowania bojowego zazwyczaj odbywa się na trasie powrotnej i uwzględnianie go jest niecelowe. Kołowanie przed startem określa się czasem pracy na ziemi i przyjmuje się wówczas zużycie paliwa 10kg/min. dla samolotów MiG-21 i 20kg/min dla samolotów MiG-29.

Modele lotu na poszczególnych etapach powinny ponadto odwzorowywać wpływ istotnych czynników zgodnie z wnioskami przedstawionymi w rozdziale 2.

### 3.2. Dane potrzebne do wykonania obliczeń

3.2.1. Zapas paliwa - całkowity zapas paliwa na samolocie  $m_{\text{pal.c}}$  oblicza się z zależności (10):

$$m_{\text{pal.c}} = (Q_{\text{pal.gł.}} + Q_{\text{pal.PZP}}) * \text{gamma}; \quad (10)$$

gdzie:

$Q_{\text{pal.gł.}}$  - eksploatacyjna pojemność zbiorników głównej instalacji paliwowej;

$Q_{\text{pal.PZP}}$  - eksploatacyjna pojemność zbiorników podwieszanych;

gamma - gęstość paliwa [kg/l];

3.2.2. Gwarantowany techniczny zapas paliwa  $m_t$  [kg] stanowi:

$$m_t = 0.07 * m_{\text{pal.c}}; \quad (11)$$

Zapasy ten powinien być uwzględniany w obliczeniach jako rezerwa na możliwe błędy charakterystyk zużycia paliwa, powstałe w wyniku tolerancji w regulacji silnika i agregatów instalacji, zmiany aerodynamiki samolotu podczas eksploatacji.

3.2.3. Zapas paliwa na lot w ugrupowaniu - 3% paliwa całkowitego dla klucza i 7% dla eskadry (współczynnik  $K_u = \{0 \text{ lub } 0.03 \text{ lub } 0.07\}$ );

3.2.4. Zapas nawigacyjny paliwa - (7 lub 10)% paliwa całkowitego współczynnik  $K_n = \{0.07 \text{ lub } 0.1\}$ .

3.2.5. Przedstartowa masa samolotu  $m_{\text{pst.}}$

$$m_{\text{pst.}} = m_{\text{oo}} + m_{\text{pal.c}} + m_{\text{p.zew.}}; \quad (12)$$

gdzie:

$m_{\text{oo}}$  - masa pustego samolotu z uwzględnieniem paliwa "martwego";

$m_{\text{p.zew.}}$  - masa podwieszonych zewnętrznych [kg]

(uzbrojenia, belek, wyrzutni, uchwytów,  
pustych zbiorników podwieszanych,

### 3.2.6. Masa startowa $m_{st}$ .

$$m_{st.} = m_{pst.} - Qz * Tz; \quad (13)$$

gdzie:

Qz- paliwo pracy na ziemi-10kg/min dla samolotu  
MiG-21, 20kg/min dla MiG-29;

Tz - czas pracy na ziemi [min].

### 3.2.7. Sumaryczny WOC

$$WOC = WOC_s + WOC_{p.zew}; \quad (14)$$

gdzie:

$WOC_s$  - wsp. oporów czołowych samolotu bez  
podwieszeń;

$WOC_{p.zew}$  - wsp. oporów czołowych podwieszeń  
zewnętrznych.

3.2.8. Dane dodatkowe dotyczące: masy i WOC zrzucanego uzbrojenia  
-( $\mu$ ,  $WOC_u$ ), pojemności zbiorników podwieszanych (PZ) ich masy  
i WOC ( $m_z$ ,  $WOC_z$ ), minutowego zużycia paliwa podczas lądowania  
(35kg/min), zapasu paliwa na dodatkowy krąg nad lotniskiem, są  
danymi instrukcyjnymi.

### 3.2.10. Dane dotyczące parametrów zewnętrznych :

-średnia wysokość bezwzględna rejonu lotów (lotniska) -H1 [m]  
(przyjęto ograniczenie od dołu 0m n.p.m., a od góry 3000m  
n.p.m.);

-średnia temperatura w rejonie lotów [stC] (przyjęto ograniczenie  
od 0 do +30°C ze względu na brak danych dla samolotu  
MiG-21Bis);

3.2.11. Paliwo walki powietrznej - ustala użytkownik;

### 3.3. Charakterystyka startu i lotu wznoszącego

Określenie czasu, paliwa i drogi startu z naborem określonej wysokości H dokonuje się postępując zgodnie z "KLUCZEM 2" przedstawionym na rysunkach 2.6.1, 2.6.2, 2.6.3 w odniesieniu do charakterystyk tam zawartych [3] s.40,41,42. dla samolotu MiG-29 i analogicznie rysunkach 1 i 2 [4] s.13,14 dla samolotu MiG -21Bis.

W procedurze programowej "STREFA" będącej przedmiotem opracowania omawiane charakterystyki zostały aproksymowane wielomianami wyższych stopni metodą Lagrange'a (przy pomocy specjalnie opracowanego do tych celów programu komputerowego).

Ponadto dla zachowania ciągłości, dla parametrów wielomianów odpowiadających wartościom pośrednim między sąsiednimi charakterystykami dokonano również interpolacji.

Zrealizowanie procedur: **wznoszenie29, (wznoszenie21b)** dla danych: mst, woc, H oraz temperatury jest równoważne kalkulacjom przeprowadzonym na omawianych charakterystykach lotno-taktycznych [3],[4] według "KLUCZA 2". Dokładne postacie wymienionych procedur zawiera załącznik 2.

Należy podkreślić, że procedura **wznoszenie29 jest kopią (nieco uproszczoną** w zakresie problematyki zrzutu zbiornika dodatkowego) **procedury wznoszenie1 z programu "MRWW-29VA" [5] jak z resztą większość pozostałych procedur.** Stąd dokładne ich opisywanie, biorąc pod uwagę pozycję [5] [2] wydaje się niecelowe. Procedury dla samolotu MiG-21Bis powstały na podobnych zasadach,

pewne różnice wynikają z różnych konwencji przedstawiania charakterystyk lotno-taktycznych tych samolotów w pozycjach [3] i [4]. Na przykład dla samolotu MiG-29 zazwyczaj parametrem jest prędkość Macha, natomiast dla samolotu MiG-21 prędkość przyrzadowa.

#### 3.4. Charakterystyka lotu poziomego z ustaloną prędkością

Lot poziomy może być wykonywany z nakazaną prędkością lotu lub z prędkością maksymalnego zasięgu, której wartość trzeba określać w zależności od bieżącej masy samolotu, WOC, wysokości barometrycznej lotu, posługując się odpowiednimi charakterystykami z pozycji [3] (rys.2.7.2. s.52) lub [4] (rys.7 s.18).

W procedurze programowej "STREFA" omawiane charakterystyki zostały aproksymowane wielomianami. Ponadto dla zachowania ciągłości, w przypadku pojawienia się wysokości i WOC o wartościach pośrednich niż te, dla których określono postacie algebraiczne (w oparciu o sporządzone wykresy [3],[4]), przewidziano interpolację wartości uzyskiwanych z obliczeń dla sąsiednich, wielomianów odpowiadających istniejącym charakterystykom (podobnie jak opisano to przy omawianiu realizacji wznoszenia).

Zasada ta stosowana była konsekwentnie podczas zamiany form graficznych zawartych w pozycji [3],[4] na postacie algebraiczne wygodne w procesie programowania. Określanie prędkości maksymalnego zasięgu  $V_{mk}$  [Ma] realizuje procedura `maksymalnyzasieg29` dla samolotu MiG-29 i procedura

**maksymalny zasięg** dla samolotu MiG-21 i gdzie  $V_{mk}$  - jest właśnie prędkością maksymalnego zasięgu [Ma].

Lot poziomy zrealizowano w procedurze programowej "STREFA" tak, że po wprowadzeniu wielkości wejściowych (dotyczących długości odcinka lotu  $S$  [km], kąta skosu skrzydła, prędkości podróźnej (lub maksymalnego zasięgu) wyliczane zostaje kilometrowa zużycie paliwa  $C_k$ .

Wyliczenie wartości zużycia kilometrowego paliwa  $C_k$  w zależności od masy samolotu- $m_{st}$  [kg], prędkości Macha -  $V_{mk}$  [Ma] i wysokości barycznej lotu- $H$  [m] wykonuje się w oparciu o stosowne charakterystyki z pozycji [3],[4] postępując według przedstawionego tam "KLUCZA-1".

W procedurze "PROMIEN" określenie  $C_k$  wykonują procedury: "oblicz\_ck29", "oblicz\_ck21B".

Procedury te wykonano przy pomocy analogicznej "technologii" zamiany form graficznych na algebraiczne jak w procedurach opisanych wyżej.

Paliwo na lot poziomy ( $Q_{lp}$ ) wyliczane jest z zależności:

$$Q_{lp} = c_k \cdot S \quad [\text{kg}] \quad (15)$$

### 3.5. Charakterystyka zniżania zająćcia do lądowania

Warunki zniżania zapewniające największy zasięg przy minimalnym zużyciu paliwa przedstawione są w pozycji [3] na rysunkach 2.8.1. s.58 dla samolotu MiG-29 i w pozycji [4] na rysunku 20 s.31 dla samolotu MiG-21Bis.

Czas, paliwo i drogę zniżania określa się według przedstawionego tam "KLUCZA". Charakterystyki te. i "klucz" posługiwania się nimi odzwierciedlają procedury "znizanie29" i ,

"znizanie21b". realizowane w procedurze programowej "STREFA" dla wysokości zniżania większych od 1000m.

Dla wysokości mniejszych niż 1000m przyjęto zniżanie według procedury zajścia do lądowania z rubieży.

### 3.6. Charakterystyka manewru w płaszczyźnie poziomej

Parametry manewru kursem lotu (promień - Rz, czas zakrętu -T, paliwo manewru -Q) zależą od wysokości, prędkości lotu i kąta przechylenia (B) utrzymywanego podczas wykonywania tej figury pilotażu.

$$Rz = V^2 / g \cdot \operatorname{tg}(B); \quad [m] \quad (16)$$

gdzie: V-prędkość [m/s];

g- przyspieszenie ziemskie [m/s<sup>2</sup>].

$$T_m = 2 \cdot \Pi \cdot Rz / V; \quad (17)$$

Paliwo i czas manewru określono poprzez realizację procedur "manewr29" dla samolotu MiG29 i "manewr21" dla samolotu MiG-21Bis dla kąta stałego kąta przechylenia  $B = 30^\circ$ . Procedury te odzwierciedlają sposób posługiwania się charakterystykami z pozycji [3] (rys. 2.10.3 s.70) i pozycji [4] (rys.18 s.30 i rys.60 s.74)

### 3.7 Bilans paliwa - pozostałość paliwa dysponowana na dyżurowanie

W ogólnym przypadku bilans paliwa sporządza się obliczając zapasy paliwa na:

- START I WZNOŚZENIE;
- LOT POZIOMY NA TRASIE DOLOTU DO STREFY;

- WZNOSENIE OD WYSOKOŚCI DYŻUROWANIA DO WYSOKOŚCI POWROTU;
- LOT POZIOMY NA TRASIE DOLOTU DO LOTNISKA;
- ZNIŻANIE I ŁADOWANIE;
- DODATKOWY KRĄG NAD LOTNISKIEM;
- WALKĘ POWIETRZNA;
- ZAPAS TECHNICZNY (7% paliwa całkowitego)
- ZAPAS NAWIGACYJNY;
- ZAPAS NA LOT W UGRUPOWANIU (uwzględniany poprzez zmniejszenie czasu dyżurowania pojedynczego samolotu);

Suma wymienionych mas paliwa odjęta od całkowitego eksploatacyjnego zapasu paliwa daje paliwo dysponowane na lot w strefie.

Zgodnie z przyjętymi założeniami (podrozdział 3.7.) średnie zużycie jednostkowe paliwa podczas dyżurowania można obliczyć z zależności:

$$ck_{\text{sr}} = \frac{240 \cdot ck + T_m \cdot ck_m}{240 + T_m} ; \quad (18)$$

Czas dyżurowania pojedynczego samolotu w oparciu o zależność:

$$T_{\text{dyż}} = 3600 \cdot Q_p / (ck_{\text{sr}} \cdot V) \quad [\text{sek}] \quad (19)$$

### 3.8. Obliczanie możliwości dyżurowania jednostek LM

Problemy określania możliwości ciągłego dyżurowania zadanej liczby samolotów i potrzebnej liczby samolotów do dyżurowania w zadanych warunkach (ciągłego w czasie nieograniczonym lub w

czasie nakazanym) w aspekcie rozważań tej kwestii, zawartych w rozdziale drugim, rozwiązano praktycznie w formie opcji-2 i opcji-3 programu "STREFA".

Program posiada trzy zasadnicze opcje:

- 1.OKREŚLANIE CZASU DYŻUROWANIA POJEDYŃCZEGO SAMOŁOTU (PARY);
- 2.OKREŚLANIE MOŻLIWOŚCI CIĄGŁEGO DYŻUROWANIA ZADANEJ LICZBY SAMOŁOTÓW;
- 3.OKREŚLANIE POTRZEBNEJ LICZBY SAMOŁOTÓW DO DYŻUROWANIA W ZADANYCH WARUNKACH

W wyniku zrealizowania opcji 2 użytkownik otrzymuje informacje o możliwości dyżurowania wyznaczonej do tego zadania liczby N samolotów kolejno, dla grup dyżurujących w strefie, w składzie od 1 do N samolotów (czas w minutach podany w kolumnie 1 - EKRAN 13 załącznik 1). Czas podany w kolumnie drugiej informuje o czasie przerwy jaka jest potrzebna do ponownego pojawienia się samolotów w strefie z możliwością ciągłego dyżurowania jak w kolumnie pierwszej. Jeżeli czas dyżurowania pojedynczego samolotu jest mniejszy od 1min to użytkownik otrzymuje informacje o niemożliwości dyżurowania w wyznaczonej strefie.

Opcja 3 zapewnia podanie potrzebnej liczby samolotów jakie należy wydzielić do dyżurowania w zadanych warunkach jeżeli czas dyżurowania pojedynczego samolotu jest większy od 1min. W przeciwnym wypadku użytkownik jest informowany o niemożliwości dyżurowania w wyznaczonej strefie.

Krytyka zależności (5) skłoniła do przyjęcia podczas opracowywania algorytmu programu modelu matematycznego, który jest opisem formalnym planowej tabeli lotów sporządzanej dla przypadków wszystkich możliwych składów, dyżurujących w powietrzu grup. Założono, że niezależnie od składu grupy w ugrupowaniu zwartym nie wystąpi więcej niż klucz samolotów. Skalkulowano ponadto czasy potrzebne na lądowanie w odstępach czasowych co 20 sekund oraz czasy potrzebne na odtworzenie gotowości bojowej po wylądowaniu, zakładając jedynie potrzebę uzupełnienia paliwa i gazów technicznych, a także osiągnięcie gotowości do startu.

Przewidziano także możliwość zmiany zadeklarowanych a priori wartości czasów odtwarzania gotowości bojowej w myśl zasady programowania globalnego z wariantowaniem zmian danych bazowych.

Czas dyżurowania ( $T_{x1}$ ) w strefie grupy w składzie  $s$  - samolotów znając czas dyżurowania pojedynczego samolotu (pary) -  $T_x$  (OPCJA-2), można obliczyć w przybliżeniu z zależności:

$$\text{Dla } s \leq 2 \quad T_{x1} = T_x; \quad (20)$$

$$\text{dla } s > 2 \quad T_{x1} = 0.97 \cdot T_x; \quad (21)$$

Czas ciągłego dyżurowania ( $T$ ) grupy w składzie  $s$ -samolotów w strefie dyżurowania można obliczyć korzystając z zależności:

$$T = N_g \cdot T_{x1}; \quad (22)$$

pod warunkiem że:

$$T_0 < (N_g - 1) \cdot T_{x1} - 2 \cdot T_D + 4 - (s - 1) / 3; \quad (23)$$

w przeciwnym wypadku wyznaczona grupa  $N$  - samolotów ma w założonych warunkach możliwość ciągłego dyżurowania w czasie nieograniczonym.

gdzie:  $N_g = N/s$  liczba grup w składzie  $s$  samolotów,

$N$  - ogólna liczba samolotów wyznaczona do dyżurowania.

Czas przerwy potrzebnej ( $T_p$ ) na uzyskanie zdolności do powtórnego utrzymywania w strefie samolotów w składzie  $s$  samolotów przez czas  $T_{x1}$  można obliczyć z zależności:

$$T_p = T_0 - (N_g - 1) \cdot T_{x1} + 2 \cdot T_D + 4 + (s - 1) / 3; \quad (24)$$

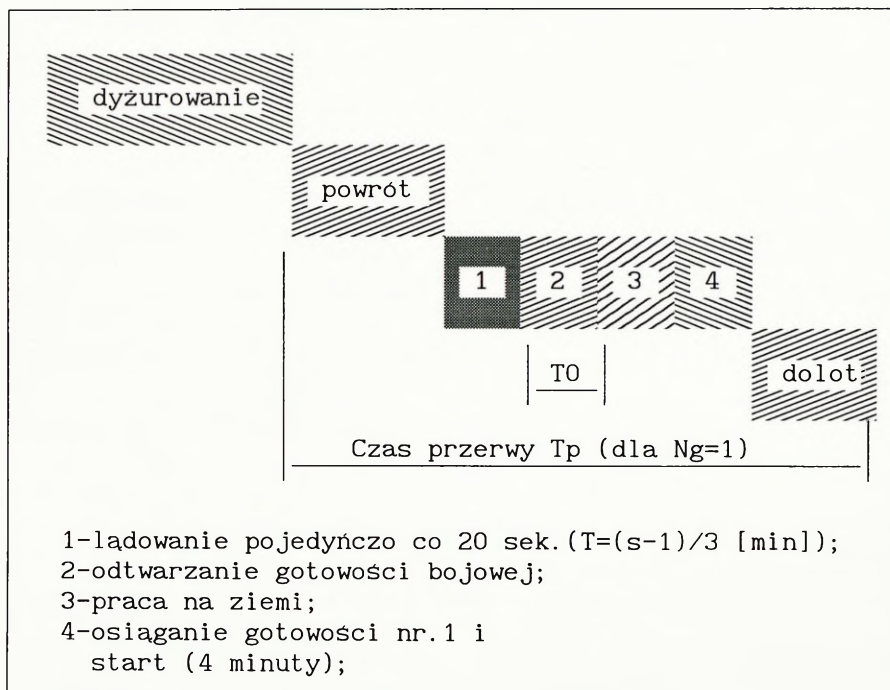
gdzie:  $T_0$  - czas odtwarzania gotowości grupy w składzie  $s$ -samolotów, dla  $s \leq 2$   $T_0 = T_1$ , dla  $s > 2$

$T_0 = ((T_3 - T_2) \cdot (s - 4) / 14) + T_2$  przyczym  $T_1$  to czas odtwarzania gotowości bojowej pary, aczasy  $T_2$  i  $T_3$  to czasy odtwarzania gotowości odpowiednio

klucza i eskadry [min] ;

2•TD - czas dolotu do strefy dyżurowania i powrotu,  
czas pracy na ziemi ( próba silnika, kołowanie )  
[min];

Cykl czynności dla jednej grupy samolotów przedstawiono na  
rysunku 3.



Rys.3. Bilans czasu dla pojedynczej grupy samolotów  
biorących udział w dyżurowaniu.

Jeżeli wielkością poszukiwaną jest ogólna liczba samolotów  
( $N_1$ ), potrzebna do realizacji zadania utrzymywania w strefie  
dyżurowania w czasie  $TT$  (lub ciągle - w nieograniczonym czasie)  
grupy w składzie  $s$  - samolotów (OPCJA-3), to kalkulacje należy  
przeprowadzić korzystając z zależności:

$$N = s \cdot N_g; \quad (25)$$

gdzie:

$$Ng = \max\{ [(T0+2 \cdot Td+4+(S-1)/3)/Tx1]+1; TT \cdot S/Tx1; \} \quad (26)$$

#### 4. OGÓLNY ALGORYTM OBLICZANIA MOŻLIWOŚCI DYŻUROWANIA

W celu odwzorowania zjawisk na poszczególnych modelowanych etapach postępuje się według ogólnego algorytmu polegającego na:

START



1. określeniu parametrów samolotu (masa, WOC, pozostałości paliwa) w locie do strefy dyżurowania - etapy startu i wznoszenia, lotu poziomego, zniżania do wysokości dyżurowania. Jest to możliwe ponieważ znana jest odległość i warunki lotu.



2. określeniu parametrów samolotu (masa, WOC, pozostałości paliwa) w locie powrotnym. Jest to możliwe ponieważ znana jest odległość i warunki lotu oraz masa samolotu i pozostałość paliwa po wylądowaniu.



3. określeniu pozostałości paliwa na dyżurowanie -  $Q_p$ .



JEŻELI  $Q_p < Q_{GRANICZNEGO}$ ;

TAK

NIE

"NIEMOŻLIWE DYŻUROWANI W  
ZAŁOŻONYCH WARUNKACH"

koniec

↓

Wyznaczenie średniej masy samolotu  
w strefie podczas dyżurowania

↓

Wyznaczenie średniego jednostkowego  
zużycia paliwa  $ck_{sr}$  podczas  
dyżurowania:

- wyznaczenie średniego zużycia  
jednostkowego w locie poziomym  
 $ck$ ;
- wyznaczenie czasu manewru o 360  
stopni  $T_m$  [sek] i jednostkowego  
zużycia paliwa podczas manewru  
 $ck_m$

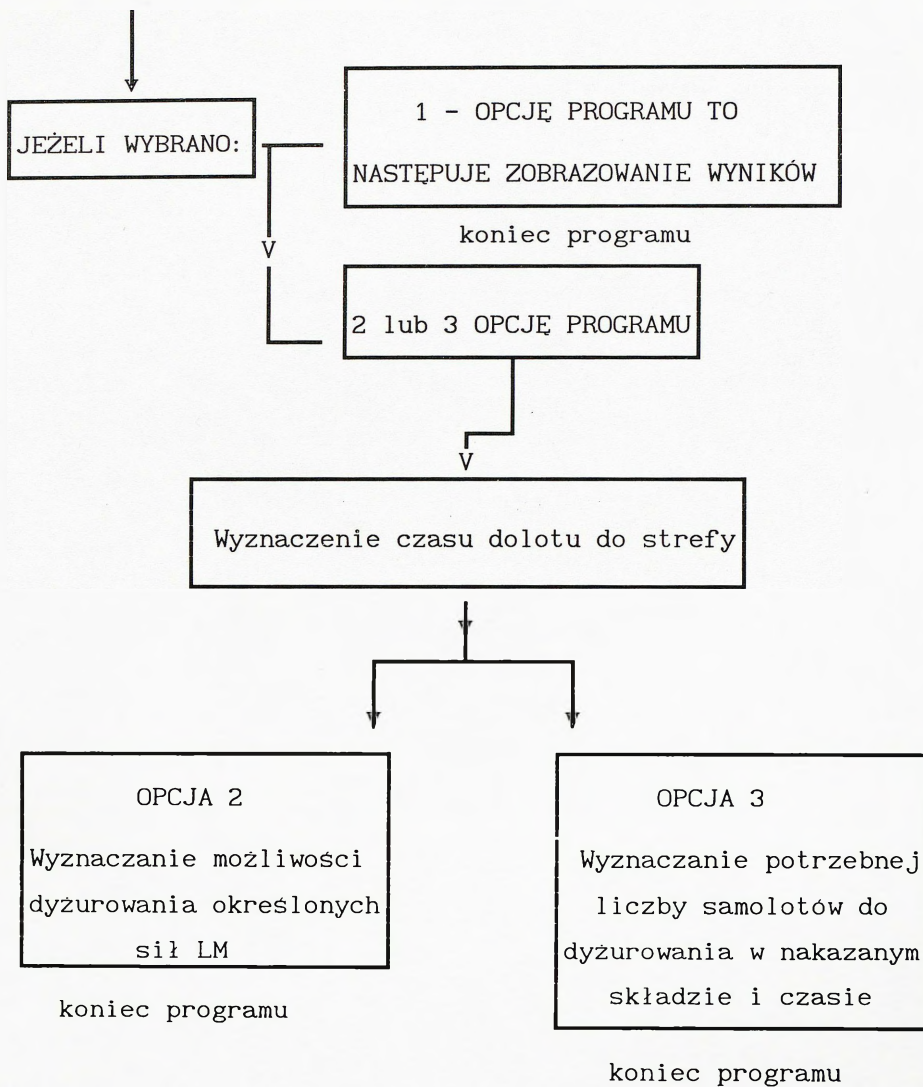
$$ck_{sr} = \frac{240 \cdot ck + T_m \cdot ck_m}{240 + T_m}$$

↓

Wyznaczenie czasu dyżurowania  
pojedynczego samolotu, pary

$$T_{dyz} = 3600 \cdot Q_p / (ck_{sr} \cdot V) \quad [\text{sek}]$$

↓



## 5. OPIS UŻYTKOWANIA MODUŁU PROGRAMOWEGO "STREFA"

### I PRZYKŁAD OBLICZEŃ

#### 5.1. Podstawowe wiadomości o programie

Przeznaczeniem programu "STREFA" jest określanie możliwości dyżerowania w powietrzu w wyznaczonych strefach dyżerowania samolotów MiG-21Bis i MiG-29, z możliwością modelowania profilu dolotu i powrotu do strefy jak i warunków dyżerowania w niej.

Program posiada trzy zasadnicze opcje:

1. OKREŚLANIE CZASU DYŻEROWANIA POJEDYŃCZEGO SAMOLOTU (PARY);
2. OKREŚLANIE MOŻLIWOŚCI CIĄGŁEGO DYŻEROWANIA ZADANEJ LICZBY SAMOLOTÓW;
3. OKREŚLANIE POTRZEBNEJ LICZBY SAMOLOTÓW DO DYŻEROWANIA W ZADANYCH WARUNKACH

Wynikiem działania opcji 1 jest czas dyżerowania pojedynczego samolotu (pary) podany w minutach jeżeli w zadeklarowanych warunkach czas ten jest większy od 1min.

W wyniku zrealizowania opcji 2 użytkownik otrzymuje informacje o możliwości dyżerowania wyznaczonej do tego zadania liczby N samolotów kolejno, dla grup dyżerujących w strefie, w składzie od 1 do N samolotów (czas w minutach podany w kolumnie 1 - EKRAN 13 załącznik 1). Czas podany w kolumnie drugiej informuje o czasie przerwy jaka jest potrzebna do ponownego pojawienia się samolotów w strefie z możliwością ciągłego dyżerowania jak w kolumnie pierwszej. Jeżeli czas dyżerowania pojedynczego samolotu jest mniejszy od 1min to użytkownik otrzymuje informacje o niemożliwości dyżerowania w wyznaczonej strefie.

Opcja 3 zapewnia podanie potrzebnej liczby samolotów jakie należy wydzielić do dyżerowania w zadanych warunkach jeżeli czas dyżerowania pojedynczego samolotu jest większy od 1min. W przeciwnym wypadku użytkownik jest informowany o niemożliwości dyżerowania w wyznaczonej strefie.

Modelowanie lotu możliwe jest w ramach przyjętych założeń i ograniczeń. Założono że:

- zakres wysokości lotu będzie się zawierał w przedziale od 200m do 9000m;

- zakres deklarowanych średnich wysokości rejonu lotów będzie się zawierał w przedziale od 0 do 3000m n.p.m.;
- zakres prędkości lotu będzie się zawierał w przedziale od prędkości maksymalnego zasięgu do prędkości rzeczywistej odpowiadającej prędkości 0.90Ma na danej wysokości lotu;
- wysokość dolotu i powrotu nie powinna być mniejsza od wysokości dyżurowania;
- zakres uwzględnianych temperatur przez algorytm programu zawiera się w granicach  $\pm 15^{\circ}\text{C}$  od temperatury standardowej (czyli od 0 do  $30^{\circ}\text{C}$ );
- maksymalna liczba samolotów dyżurująca w strefie w ugrupowaniu zwartym nie jest większa od klucza.

Ponadto przyjęto do kalkulacji w opcjach 2 i 3, czas startu i zajęcia gotowości nr.1 - 4 min. oraz warunki lądowania z rubieży (pojedynczo co 20 sekund z rozpuszczeniem do lądowania na trasie dolotu do lotniska lądowania).

Algorytm programu oparto na treściach pozycji:

- \* "SAMOLOT MiG-29 i MiG-29UB Z SILNIKAMI RD-33, Instrukcja obliczania zasięgu i długotrwałości lotu". WLOP 79/91, Poznań 1992, [3];
- \* "ИНСТРУКЦИЯ ПО РАСЧЕТУ ДАЛЬНОСТИ И ПРОДОЛЖИТЕЛЬНОСТИ ПОЛЕТА САМОЛЕТА МиГ 21БИС С ДВИГАТЕЛЕМ Р25 300";

Program "STREFA" może być wykorzystywany przy użyciu komputerów IBM PC z kartą graficzną VGA lub sVGA.

Program uruchamia się poprzez uaktywnienie zbioru strefa.exe pod warunkiem dysponowania w podkatalogu zbiorem egavga.bgi

Wprowadzanie danych odbywa się metodą interakcyjną. Użytkownik powinien dostosować się do poleceń pojawiających się na monitorze ekranowym. Przekroczenie ograniczeń jest sygnalizowane.

W wypadku wykonywania kalkulacji dla lotu w ugrupowaniu o określonym składzie rezultaty kalkulacji odnoszą się do samolotu, którego warunki lotu wymagają największego zużycia paliwa spośród statków powietrznych wykonujących lot.

Modelowanie elementów lotu rozpoczyna się od wyboru jednej z trzech opcji programu, a następnie wyboru typu samolotu myśliwskiego, wariantu ładunku bojowego oraz zadeklarowania danych wejściowych takich jak:

- gęstość paliwa (wybór przy pomocy bloku świetlnego);
- czas pracy silników na ziemi (wybór przy pomocy bloku świetlnego);
- średniej temperatury w rejonie lotów;
- średniej wysokości względnej rejonu lotów;
- paliwa walki powietrznej;
- warunków dyżurowania w strefie (wysokość, prędkość - nakazana lub maksymalnego zasięgu);
- warunków dolotu do strefy (odległość wysokość lotu i prędkość - nakazana lub maksymalnego zasięgu);
- warunków powrotu ze strefy (odległość wysokość lotu i prędkość - nakazana lub maksymalnego zasięgu);

w wypadku opcji 2 dodatkowo:

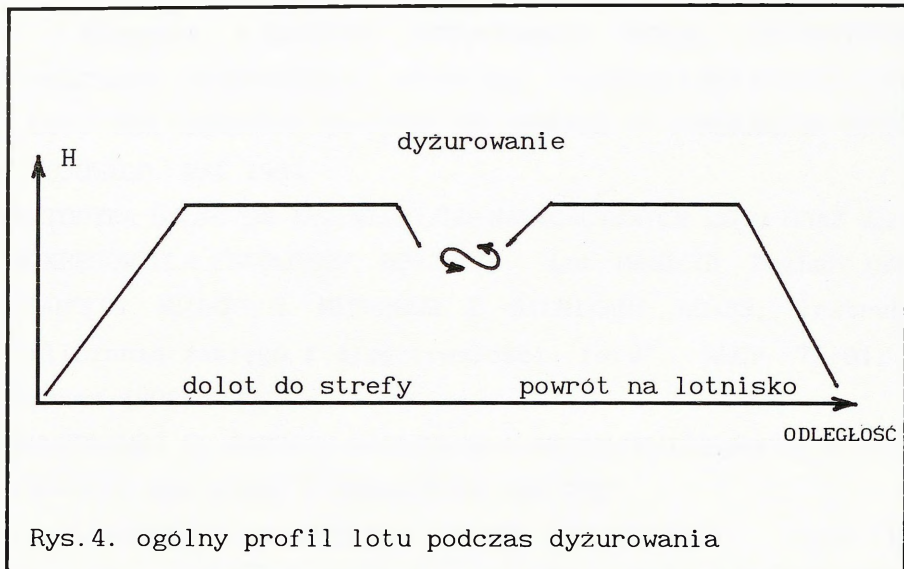
- podania ogólnej liczby samolotów wyznaczonych do dyżurowania;
- zatwierdzenia lub zmiany przyjętych czasów odtwarzania gotowości bojowej;

w wypadku opcji 3 dodatkowo:

- określenia składu dyżurującej grupy w strefie;
- podania nakazanego czasu dyżurowania ("dyżurowanie ciągle" lub podanie czasu w minutach);
- zatwierdzenia lub zmiany przyjętych czasów odtwarzania gotowości bojowej;

"BLOK ŚWIETLNY" przesuwa się przy użyciu klawiatury - "strzałek", a zatwierdzenie zadeklarowanej wartości następuje po naciśnięciu klawisza "ENTER". Naciśnięcie klawisza "ENTER" wprowadza do programu wybrane przy pomocy klawiatury alfa-numerycznej wartości liczbowe, które można zmieniać przy użyciu klawiszy "DEL" lub "BACKSPACE" do tego momentu.

Ogólny profil lotu przyjęty w OPCJI-1 przedstawia rysunek 4.



### 5.2. Przykład obliczeń

Przykład obliczeń możliwości dyżurowania z wykorzystaniem procedury programowej "STREFA" przedstawiono w załączniku 1 stanowiącym treści wybranych informacji zobrazowywanych na monitorze ekranowym komputera, dotyczących wprowadzanych wielkości wejściowych, komunikatów i wyników kalkulacji.

Ekrany o numerach od 1 do 8 dotyczą OPCJI-1 programu. Ekrany o numerach 13 dotyczą OPCJI-2, a pozostałe od 14 do 19 OPCJI 3.

## BIBLIOGRAFIA

1. P. MAKOWSKI A. GRZELKA "OPRACOWANIE METOD, ALGORYTMÓW I PROCEDUR PROGRAMOWYCH OBLICZEŃ INŻYNIERYJNO-NAWIGACYJNYCH LOTU DLA SAMOLOTU Su-22M4 ZWIĄZANYCH Z WYKONANIEM ZADANIA BOJOWEGO. WAT 1994.
2. "METODYKA OBLICZEŃ INŻYNIERYJNO-NAWIGACYJNYCH LOTU ORAZ ZESTAW WYKRESÓW DLA SAMOLOTÓW BOJOWYCH. Lot.1928/79 Poznań 1980.
3. "SAMOLOT MiG-29 i MiG-29UB Z SILNIKAMI RD-33, Instrukcja obliczania zasięgu i długotrwałości lotu". WLOP 79/91, Poznań 1992;
4. "ИНСТРУКЦИЯ ПО РАСЧЕТУ ДАЛЬНОСТИ И ПРОДОЛЖИТЕЛЬНОСТИ ПОЛЕТА САМОЛЕТА МиГ 21БИС С ДВИГАТЕЛЕМ Р25 300";
5. P. MAKOWSKI A. GRZELKA "WYBÓR TRAJEKTORII LOTU NA PRZECHWYCENIE - OKREŚLANIE MOŻLIWYCH RUBIEŻY WPROWADZENIA DO WALKI". WAT 1993.

Złącznik 1

WYBRANE WYDRUKI TREŚCI ZOBRAZOWANYCH NA EKRANIE MONITORA  
PODCZAS UŻYTKOWANIA PROGRAMU "PROMIEŃ"

**OKRESLANIE:**

- czasu dyzuruwania pojedynczego samolotu
- mozliwosci ciaglego dyzuruwania zadanej liczby samolotow
- potrzebnej liczby samolotow do dyzuruwania w zadanych warunkach

**WYBIERZ WARIANT I NACISNIJ "Enter"**

Wybierz  
TYP myśliwca

- MiG-21bis**
- MiG-29



WYBIERZ WARIANT I NACISNIJ "Enter"

Wybierz  
TYP myśliwca

**MiG-21bis**

MiG-29

WARIANT UZBROJENIA  
+ komplet amunicji do działka

Bez zbiorników

z 1 x 490 L

z 1 x 765 L

**z 2 x 490 L**

z 2 x 490 L+1 x 765 L

z 3 x 490 L

WYBIERZ WARIANT I NACISNIJ "Enter"

Wybierz  
TYP myśliwca

MiG-21bis  
MiG-29

WARIANT UZBROJENIA  
+ komplet amunicji do działka

bez uzbrojenia

2 x R-3s

2 x R-3R

2 x R-60M

4 x R-60M

2 x R-13M

2 x UB-16

WYBIERZ WARIANT I NACISNIJ "Enter"

Wybierz  
TYP myśliwca

**MiG-21bis**

MiG-29

WARIANT UZBROJENIA  
+ komplet amunicji do działka

bez uzbrojenia

2 x R-3s

2 x R-3R

**2 x R-60M**

4 x R-60M

2 x R-13M

2 x UB-16

GĘSTOŚĆ  
PALIWA

0.755

0.765

0.775

0.785

0.795

**0.805**

0.815

0.825

0.835

0.845

Czas pracy  
na ziemi [ min. ]

5

6

**7**

8

9

10

WYBIERZ WARIANT I NACISNIJ "Enter"

Wybierz  
TYP myśliwca

**MiG-21bis**  
MiG-29

WARIANT UZBROJENIA  
+ komplet amunicji do działka

bez uzbrojenia  
2 x R-3s  
2 x R-3R  
**2 x R-60M**  
4 x R-60M  
2 x R-13M  
2 x UB-16

GĘSTOŚĆ  
PALIWA

0.755  
0.765  
0.775  
0.785  
0.795  
**0.805**  
0.815  
0.825  
0.835  
0.845

Czas pracy  
na ziemi [ min. ]

5  
6  
**7**  
8  
9  
10

Zapewnia nawigacyjny

**7 %**  
10 %

WYBIERZ WARIANT I NACISNIJ "Enter"

| DANE   | PRZYJETE  | DO | OBLICZEN |
|--|-----------|----|----------|
| Typ myśliwca . . . *                                 | MiG-21bis |    |          |
| Wariant uzbrojenia .. *                              | 2 x R-60M |    |          |
| Gęstość paliwa . . . . . *                           |           |    | 0.805    |
| Czas pracy na ziemi . . . . . [min.].. *             |           |    | 7        |
| Zapasy nawigacyjny . . . . . [ % ]... *              |           |    | 7 1/2    |
| Srednia temp. w rejonie lotow [stop. C ]... *        |           |    | 24       |
| Srednia wys.rej.lotow (lotniska) n.p.m [m].. *       |           |    |          |
| Paliwo walki powietrznej . . . . . [kg]... *         |           |    |          |
| Wysokosc dyżurowania . . . . . [m]... *              |           |    |          |
| Prędkosc dyżurowania . . . . . [km/h]... *           |           |    |          |
| Odleglosc strefy dyżurowania . . . . . [km]... *     |           |    |          |
| Wysokosc dolotu do strefy . . . . . [m]... *         |           |    |          |
| Prędkosc dolotu do strefy . . . . . [km/h]... *      |           |    |          |
| Odleglosc do lotniska ladowania . . . . . [km ]... * |           |    |          |
| Wysokosc powrotu . . . . . [m]... *                  |           |    |          |
| Prędkosc powrotu . . . . . [km/h]... *               |           |    |          |

Srednia wysokosc rejonu lotow [metry]. 120

| DANE                                 | PRZYJĘTE   | DO | OBLICZEN                |
|--------------------------------------|------------|----|-------------------------|
| Typ myśliwca .....                   | MiG-21bis  |    |                         |
| Wariant uzbrojenia ..                | 2 x R-60M  |    |                         |
| Gęstość paliwa .....                 |            | *  | 0.805                   |
| Czas pracy na ziemi.....             | [min.]     | *  | 7                       |
| Zapasy nawigacyjny.....              | [ % ]      | *  | 7 ½                     |
| Średnia temp. w rejonie lotów        | [stop. C ] | *  | 24                      |
| Średnia wys.rej.lotów (lotniska)     | n.p.m [m]. | *  | 120                     |
| Paliwo walki powietrznej.....        | [kg]       | ** | 500                     |
| Wysokość dyżurowania.....            | [m]        | ** | 3000                    |
| Prędkość dyżurowania.....            | [km/h]     | ** | 732 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....    | [km]       | ** | 80                      |
| Wysokość dolotu do strefy.....       | [m]        | ** | 5000                    |
| Prędkość dolotu do strefy.....       | [km/h]     | ** | 800 ( U nakazana )      |
| Odległość do lotniska lądowania..... | [km ]      | ** | 50                      |
| Wysokość powrotu.....                | [m]        | ** | 5000                    |
| Prędkość powrotu.....                | [km/h]     | ** |                         |

Wybierz rodzaj prędkości powrotu

**U maksymalnego zasięgu**

U nakazana

| DANE  | PRZYJETE  | DO | OBLICZEN                |
|---|-----------|----|-------------------------|
| Typ myśliwca .....                          | MiG-21bis |    |                         |
| Wariant uzbrojenia ..                       | 2 x R-60M |    |                         |
| Gęstość paliwa .....                        |           | *  | 0.805                   |
| Czas pracy na ziemi.....                    | [min.]..  | *  | 7                       |
| Zapasy nawigacyjny.....                     | [ % ]..   | *  | 7 %                     |
| Średnia temp. w rejonie lotów [stop. C ]..  |           | *  | 24                      |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]. |           | *  | 120                     |
| Paliwo walki powietrznej.....               | [kg]..    | *  | 500                     |
| Wysokość dyżurowania.....                   | [m]..     | *  | 3000                    |
| Prędkość dyżurowania.....                   | [km/h] .. | *  | 732 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....           | [km]..    | *  | 80                      |
| Wysokość dolotu do strefy.....              | [m]..     | *  | 5000                    |
| Prędkość dolotu do strefy.....              | [km/h] .. | *  | 800 ( U nakazana )      |
| Odległość do lotniska lądowania.....        | [km ]..   | *  | 50                      |
| Wysokość powrotu.....                       | [m]..     | *  | 5000                    |
| Prędkość powrotu.....                       | [km/h] .. | *  | 778 ( U maks. zasięgu ) |

Czas dyżurowania pojedynczego samolotu MiG-21bis = 113 min

**OKRESLANIE:**

- czasu dyzuruwania pojedynczego samolotu
- mozliwosci ciaglego dyzuruwania zadanej liczby samolotow
- potrzebnej liczby samolotow do dyzuruwania w zadanych warunkach

**WYBIERZ WARIANT I NACISNIJ "Enter"**

| DANE  | PRZYJETE              | DO | OBLICZEN                |
|---|-----------------------|----|-------------------------|
| Typ myśliwca .....                              | * MiG-29              |    |                         |
| Wariant uzbrojenia ..                           | * 4 x R-73 + 2 x R-60 |    |                         |
| Gęstość paliwa .....                            |                       | *  | 0.795                   |
| Czas pracy na ziemi.....                        | [min.]                | *  | 8                       |
| Zapasy nawigacyjny.....                         | [ % ]                 | *  | 10 %                    |
| Średnia temp. w rejonie lotów [stop. C ].....   |                       | *  | 15                      |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]..... |                       | *  | 120                     |
| Paliwo walki powietrznej.....                   | [kg]                  | *  | 500                     |
| Wysokość dyżurowania.....                       | [m]                   | *  | 3500                    |
| Prędkość dyżurowania.....                       | [km/h]                | *  | 648 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....               | [km]                  | *  | 90                      |
| Wysokość dolotu do strefy.....                  | [m]                   | *  | 4000                    |
| Prędkość dolotu do strefy...                    | [km/h]                | *  | 800 ( U nakazana )      |
| Wysokość powrotu...                             | [m]                   | *  | 5000                    |
| Prędkość powrotu.....                           | [km/h]                | *  | 665 ( U maks. zasięgu ) |
| Ogólna liczba samolotów w dyżurowaniu.....      | *                     |    |                         |

Ogólna liczba samolotów w dyżurowaniu. ... 12

| DANE                                       | PRZYJETE              | DO  | OBLICZEN                |
|--|-----------------------|-----|-------------------------|
| Typ myśliwca .....                         | * MiG-29              |     |                         |
| Wariant uzbrojenia ..                      | * 4 x R-73 + 2 x R-60 |     |                         |
| Gęstość paliwa .....                       |                       | *   | 0.795                   |
| Czas pracy na ziemi.....                   | [min.]..              | *   | 8                       |
| Zapasy nawigacyjny.....                    | [ % ]..               | *   | 10 %                    |
| Średnia temp. w rejonie lotów              | [stop. C ]..          | *   | 15                      |
| Średnia wys.rej.lotów (lotniska)           | n.p.m [m].            | *   | 120                     |
| Paliwo walki powietrznej.....              | [kg]..                | ..* | 500                     |
| Wysokość dyżurowania.....                  | [m]..                 | ..* | 3500                    |
| Prędkość dyżurowania.....                  | [km/h]                | ..* | 648 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....          | [km]..                | ..* | 90                      |
| Wysokość dolotu do strefy.....             | [m]..                 | ..* | 4000                    |
| Prędkość dolotu do strefy...               | [km/h]                | ..* | 800 ( U nakazana )      |
| Wysokość powrotu.....                      | [m].....              | ..* | 5000                    |
| Prędkość powrotu.....                      | [km/h].....           | ..* | 665 ( U maks. zasięgu ) |
| Ogólna liczba samolotów w dyżurowaniu..... | * 12                  |     |                         |

CZASY ODTWARZANIA GOTOWOSCI BOJOWEJ

Czy zmieniasz dane ?

|         | MiG-21bis |         | MiG-29 |         |
|---------|-----------|---------|--------|---------|
|         | Ze zb.    | Bez zb. | Ze zb. | Bez zb. |
| Para    | 25        | 20      | 25     | 20      |
| Klucz   | 35        | 30      | 35     | 30      |
| Eskadra | 90        | 60      | 90     | 60      |

|     |
|-----|
| Tak |
| Nie |

| DANE  | PRZYJETE            | DO | OBLICZEN |
|---|---------------------|----|----------|
| Typ myśliwca .....                          | MiG-29              |    |          |
| Wariant uzbrojenia ..*                      | 4 x R-73 + 2 x R-60 |    |          |
| Gęstość paliwa .....                        |                     | *  | 0.795    |
| Czas pracy na ziemi.....                    | [min.]              | *  | 8        |
| Zapasy nawigacyjny.....                     | [ % ]               | *  | 10 %     |
| Średnia temp. w rejonie lotów [stop. C ]..  |                     | *  | 15       |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]. |                     | *  | 120      |
| Paliwo walki powietrznej.....               | [kg]                | *  | 500      |
| Wysokość dyżurowania.....                   | [m]                 | *  | 3500     |
| Prędkość dyżurowania.....                   | [km/h]              | *  | 648      |
| Odległość strefy dyżurowania.....           | [km]                | *  | 90       |
| Wysokość dolotu do strefy.....              | [m]                 | *  | 4000     |
| Prędkość dolotu do strefy.....              | [km/h]              | *  | 800      |
| Wysokość powrotu.....                       | [m]                 | *  | 5000     |
| Prędkość powrotu.....                       | [km/h]              | *  | 665      |
| Ogólna liczba samolotów w dyżurowaniu.....  |                     | *  | 12       |

| S  | Czas        | Czas    |
|----|-------------|---------|
| K  | ciągł.      | przerwy |
| L  | dyżur.      |         |
| A  | [ min ]     | [ min ] |
| D  |             |         |
| 1  | Dyż. ciągłe |         |
| 2  | Dyż. ciągłe |         |
| 3  | 133.7       | 7.71    |
| 4  | 100.3       | 43.62   |
| 5  | 66.87       | 79.54   |
| 6  | 66.87       | 82.01   |
| 7  | 33.44       | 117.9   |
| 8  | 33.44       | 120.4   |
| 9  | 33.44       | 122.8   |
| 10 | 33.44       | 125.3   |
| 11 | 33.44       | 127.8   |
| 12 | 33.44       | 130.3   |

CZASY ODTWARZANIA GOTOWOSCI BOJOWEJ

|         | MiG-21bis |         | MiG-29 |         |
|---------|-----------|---------|--------|---------|
|         | Ze zb.    | Bez zb. | Ze zb. | Bez zb. |
| Para    | 25        | 20      | 25     | 20      |
| Klucz   | 35        | 30      | 35     | 30      |
| Eskadra | 90        | 60      | 90     | 60      |

**OKRESLANIE:**

- czasu dyzowania pojedynczego samolotu
- mozliwosci ciaglego dyzowania zadanej liczby samolotow
- potrzebnej liczby samolotow do dyzowania w zadanych warunkach

**WYBIERZ WARIANT I NACISNIJ "Enter"**

| DANE   | PRZYJETE    | DO | OBLICZEN                |
|--|-------------|----|-------------------------|
| Typ myśliwca                                 | * MiG-21bis |    |                         |
| Wariant uzbrojenia                           | ** 4 x R-3S |    |                         |
| Gęstość paliwa                               |             | *  | 0.805                   |
| Czas pracy na ziemi                          | [min.]      | *  | 7                       |
| Zapasy nawigacyjny                           | [ % ]       | *  | 7 %                     |
| Średnia temp. w rejonie lotów                | [stop. C ]  | *  | 12                      |
| Średnia wys.rej.lotów (lotniska) n.p.m       | [m]         | *  | 190                     |
| Paliwo walki powietrznej                     | [kg]        | *  | 450                     |
| Wysokość dyżurowania                         | [m]         | ** | 2000                    |
| Prędkość dyżurowania                         | [km/h]      | ** | 706 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania                 | [km]        | *  | 50                      |
| Wysokość dolotu do strefy                    | [m]         | ** | 2500                    |
| Prędkość dolotu do strefy                    | [km/h]      | ** | 715 ( U maks. zasięgu ) |
| Wysokość powrotu                             | [m]         | ** | 3000                    |
| Prędkość powrotu                             | [km/h]      | ** | 725 ( U maks. zasięgu ) |
| Skład grupy dyżurującej w strefie ( N < 12 ) | *           |    |                         |

Skład grupy dyżurującej w strefie (N<12)

4

| DANE  | PRZYJETE    | DO | OBLICZEN                |
|---|-------------|----|-------------------------|
| Typ myśliwca .....                              | * MiG-21bis |    |                         |
| Wariant uzbrojenia ..*                          | 4 x R-3S    |    |                         |
| Gęstość paliwa .....                            |             | *  | 0.805                   |
| Czas pracy na ziemi.....                        | [min.]      | *  | 7                       |
| Zapasy nawigacyjny.....                         | [ % ]       | *  | 7 %                     |
| Średnia temp. w rejonie lotów [stop. C ].....   |             | *  | 12                      |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]..... |             | *  | 190                     |
| Paliwo walki powietrznej.....                   | [kg]        | *  | 450                     |
| Wysokość dyżurowania.....                       | [m]         | *  | 2000                    |
| Prędkość dyżurowania.....                       | [km/h]      | *  | 706 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....               | [km]        | *  | 50                      |
| Wysokość dołotu do strefy.....                  | [m]         | *  | 2500                    |
| Prędkość dołotu do strefy.....                  | [km/h]      | *  | 715 ( U maks. zasięgu ) |
| Wysokość powrotu.....                           | [m]         | *  | 3000                    |
| Prędkość powrotu.....                           | [km/h]      | *  | 725 ( U maks. zasięgu ) |
| Skład grupy dyżurującej w strefie ( N < 12 )    |             | *  | 4                       |

Przez jaki okres czasu masz dyżurować

Ciągle w czasie nieograniczonym  
W ciągu określonego czasu

| DANE   | PRZYJETE          | DO        | OBLICZEN            |
|--|-------------------|-----------|---------------------|
| Typ myśliwca                                 | ..... *           | MiG-21bis |                     |
| Wariant uzbrojenia                           | ..*               | 4 x R-3S  |                     |
| Gęstość paliwa                               | ..... *           | 0.805     |                     |
| Czas pracy na ziemi                          | ..... [min.] .. * | 7         |                     |
| Zapasy nawigacyjny                           | ..... [ % ] .. *  | 7 %       |                     |
| Średnia temp. w rejonie lotów                | [stop. C] .. *    | 12        |                     |
| Średnia wys. rej. lotów (lotniska) n.p.m     | [m] .. *          | 190       |                     |
| Paliwo walki powietrznej                     | ..... [kg] .. *   | 450       |                     |
| Wysokość dyżurowania                         | ..... [m] .. *    | 2000      |                     |
| Prędkość dyżurowania                         | ..... [km/h] .. * | 706       | ( U maks. zasięgu ) |
| Odległość strefy dyżurowania                 | ..... [km] .. *   | 50        |                     |
| Wysokość dolotu do strefy                    | ..... [m] .. *    | 2500      |                     |
| Prędkość dolotu do strefy                    | ..... [km/h] .. * | 715       | ( U maks. zasięgu ) |
| Wysokość powrotu                             | ..... [m] .. *    | 3000      |                     |
| Prędkość powrotu                             | ..... [km/h] .. * | 725       | ( U maks. zasięgu ) |
| Skład grupy dyżurującej w strefie ( N < 12 ) | ..... *           | 4         |                     |

Czas dyżurowania [min] 90

| DANE  | PRZYJETE  | DO | OBLICZEN                |
|---|-----------|----|-------------------------|
| Typ myśliwca .....                              | MiG-21bis |    |                         |
| Wariant uzbrojenia ..*                          | 4 x R-3S  |    |                         |
| Gęstość paliwa .....                            |           |    | 0.805                   |
| Czas pracy na ziemi.....[min.]....*             |           |    | 7                       |
| Zapasy nawigacyjny.....[ % ]....*               |           |    | 7 ½                     |
| Średnia temp. w rejonie lotów .[stop. C ]....*  |           |    | 12                      |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]....* |           |    | 190                     |
| Paliwo walki powietrznej.....[kg].....*         |           |    | 450                     |
| Wysokość dyżurowania.....[m].....*              |           |    | 2000                    |
| Prędkość dyżurowania.....[km/h]....*            |           |    | 706 ( U maks. zasięgu ) |
| Odległość strefy dyżurowania.....[km]....*      |           |    | 50                      |
| Wysokość dolotu do strefy.....[m].....*         |           |    | 2500                    |
| Prędkość dolotu do strefy.....[km/h]....*       |           |    | 715 ( U maks. zasięgu ) |
| Wysokość powrotu.....[m].....*                  |           |    | 3000                    |
| Prędkość powrotu.....[km/h]....*                |           |    | 725 ( U maks. zasięgu ) |
| Skład grupy dyżurującej w strefie ( N < 12 ) *  |           |    | 4                       |
| Dyżurowanie w czasie [min].....*                |           |    | 90                      |

Sredni czas odtwarzania gotowosci 4 samolotow  
wynosi 35 min

Czy zmieniasz czas  
odtworzenia got.boj. ?

Tak

Nie

| DANE  | PRZYJĘTE  | DO | OBLICZEN |
|---|-----------|----|----------|
| Typ myśliwca .....                              | MiG-21bis |    |          |
| Wariant uzbrojenia ..*                          | 4 x R-3S  |    |          |
| Gęstość paliwa .....                            |           |    | 0.805    |
| Czas pracy na ziemi.....                        | [min.]    |    | 7        |
| Zapasy nawigacyjny.....                         | [ % ]     |    | 7 %      |
| Średnia temp. w rejonie lotów [stop. C ]....*   |           |    | 12       |
| Średnia wys.rej.lotów (lotniska) n.p.m [m]....* |           |    | 190      |
| Paliwo walki powietrznej.....                   | [kg]      |    | 450      |
| Wysokość dyżurowania.....                       | [m]       |    | 2000     |
| Prędkość dyżurowania.....                       | [km/h]    |    | 706 (    |
| Odległość strefy dyżurowania.....               | [km]      |    | 50       |
| Wysokość dolotu do strefy.....                  | [m]       |    | 2500     |
| Prędkość dolotu do strefy.....                  | [km/h]    |    | 715 (    |
| Wysokość powrotu.....                           | [m]       |    | 3000     |
| Prędkość powrotu.....                           | [km/h]    |    | 725 (    |
| Skład grupy dyżurującej w strefie ( N < 12 ) *  |           |    | 4        |
| Dyżurowanie w czasie [min].....*                |           |    | 90       |

Dla zapewnienia nakaza warunków dyżurowania potrzeba nie mniej niż 12 samolotów

Średni czas odtwarzania gotowości 4 samolotów wynosi 35 min

## Załącznik 2

### WERSJA ŹRÓDŁOWA PROGRAMU "STREFA" - WYLISTOWANE PROCEDURY PROGRAMOWE (ROZSZERZENIE "PAS")

Procedura programowa "STREFA" w wersji źródłowej składa się z unitów:

1. dyzur\_n.pas;
2. stary.pas;
3. danesys.pas;
4. dyz\_21.pas;
5. dyz\_29.pas;
6. pocz.pas;
7. typy\_n.pas;

Tor główny programu stanowi unit "dyzur\_n.pas", opcje 2 i 3 programu zawarte są w unicie "stary.pas". Procedury dotyczące charakterystyk lotno-taktycznych samolotów MiG-21Bis i MiG-29, stanowiące swoistą bazę danych, zawarto odpowiednio w unitach: "dyz\_21.pas", i "dyz\_29.pas".

## WYKAZ ZASADNICZYCH PROCEDUR I ICH FUNKCJE:

1. "oblicz\_ck21B" i "oblicz\_ck29" dokonuje obliczania zużycia kilometrowego paliwa. (ck) dla samolotu o danej masie (mst), współczynnika oporów czołowych (WOC), podczas lotu na danej wysokości przyrzadowej (H1) z prędkością przyrzadową (Vp) lub odpowiednio Macha (Vmkk).
2. "wznoszenie21B" i "wznoszenie29" - określają drogę i czas potrzebny na nabór wysokości (Swzn, Twzn) od wysokości początkowej (Hp) do końcowej mniejszej niż 10000m (Hkk) oraz zużycie paliwa w czasie wznoszenia (Qwzn) przy pracy silników na zakresie "MAKSYMALNE OBROTY" z uwzględnieniem wyjściowej masy samolotu (mst), współczynnika oporów czołowych (WOC), danej różnicy temperatury otoczenia (TE) od temperatury standardowej. Ponadto procedura określa nowe wartości zmiennych mst, WOC oraz zmiennych określających pozostałość paliwa (Qp).
3. "znizanie21B" i "znizanie29" - określają drogę i zużycie paliwa oraz czas (Szn, Qzn, Tzn) podczas ekonomicznego zniżania od wysokości wyjścia z walki (Hk) do wysokości systemu lądowania (1000m);
4. "maksymalnyzasieg21b" i "maksymalnyzasieg29" - określają prędkość maksymalnego zasięgu na danej wysokości lotu samolotów MiG21 i Mig-29 o określonym WOC lub określonej masie (mst) na danej przyrzadowej wysokości lotu (H1).
5. "manewr21B" i "manewr29" określają paliwo jednostkowe kilometrowe zużycie paliwa (ck) podczas manewru ustalonego z prędkością Vp lub Vmkk na wysokości H1 z kątem przechylenia  $B=30^\circ$ , samolotu o masie - mst i współczynnika oporów czołowych - WOC, a także określają czas manewru o  $360^\circ$  (Tm) w sekundach.

WYKAZ NAJWAŻNIEJSZYCH NIE ZDEFINIOWANYCH DOTYCHCZAS ZMIENNYCH:

H11, - nakazana wysokość rzeczywista dolotu do strefy [m];

H22, - nakazana wysokość rzeczywista dyżurowania w strefie [m];

H11, - nakazana wysokość rzeczywista powrotu na lotnisko [m];

H1 - średnia wysokość rejonu lotów n.p.m. [m];

Qw - paliwo walki powietrznej [kg];

V1, V1M, V1P, V1mz - prędkości, odpowiednio: rzeczywista, Macha, przyrządowa, maksymalnego zasięgu, na trasie dolotu do strefy;

V2, V2M, V2P, V2mz - prędkości, odpowiednio: rzeczywista, Macha, przyrządowa, maksymalnego zasięgu, podczas dyżurowania;

V3, V3M, V3P, V3mz - prędkości, odpowiednio: rzeczywista, Macha, przyrządowa, maksymalnego zasięgu, na trasie dolotu do lotniska;

Vmax- prędkość maksymalna - 0.9Ma dla bieżących warunków lotu;

Vp - bieżąca prędkość przyrządowa;

S1 - odległość strefy od lotniska startu [km];

S3 - odległość strefy od lotniska lądowania [km];

Ta1 - czas dolotu do strefy [s];

zn - zapas nawigacyjny (7% lub 10% );

Ta3 - czas dolotu ze strefy do lotniska [s];

TEL - średnia temperatura na powierzchni Ziemi w rejonie lotów;

TEM, TEM1, TEM2, TEM3, - temperatura na wysokości odpowiednio:

H1, H11, H22, H33 w stopniach Kelvina;

WOCu, WOCz, - współczynniki oporów czołowych uzbrojenia i zbiorników dodatkowych;

mu, mz, - masy uzbrojenia i zbiorników dodatkowych;

Qz - paliwo pracy na ziemi;

Tlad, Qlad, - czas i paliwo lądowania z rubieży z H=2000m;

Szn, Tzn, Qzn , Swzn, Twzn, Qwzn - drogi, czasy i paliwo zniżania  
i wznoszenia;

Tdyz1 - czas dyżurowania pojedynczego samolotu w minutach;

TD - połowa sumy czsów dolotu, powrotu i pracy na ziemi;

s - skład grupy dyżurującej jednocześnie;

Ng - liczba grup o składzie s;

N - ogólna liczba samolotów wyznaczona lub potrzebna do  
dyżurowania;

TT - nakazany czas ciągłego dyżurowania;

T1, T2, T3, - czasy odtwarzania gotowości bojowej [min] dla  
pary, klucza, eskadry;

T0 - czas odtwarzania gotowości bojowej grupy s samolotów;

Tp - czas przerwy w dyżurowaniu potrzebnej na uzyskanie zdolności  
do utrzymywania w strefie s- samolotów przez czas Tx1;

```

program dyzur_n; {24.10.1994 nowy program}
uses
typy_n, pocz, danesys, dyz_29, dyz_21,
stary, dos, crt, graph;
var
driver, mode: integer;
r: char;
sTdyz1: string[6];
Vmax, zn, Qpp, mstp: real;
flaga_ogr, ilzapas, ilczasna_z, ilgestosc,
ilwariant, iltypm: integer;
{zmienne: array[1..16] of string[30];}
Qwmax, Qw, km, V1R, H2, V2R, H3, V3R: integer;
sVmax, stel1, stel: string[6];
skoki, wspy_bar, wspy_x, wspy_y: integer;
label
kkkk, start, poczatek, w11, w10, w9,
w8, w7, w6, w4, w5, w3, w2, w1, 11;
type

tkom20=array[1..20] of string[10];
tkom30=array[1..30] of string[30];
tkom=array[1..32] of string[53];

const
t:tkom=('1','2','3','4','','','','','','','','',
'', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '',
zapas:tkom20=(' 7 %', ' 10 %', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '');
Tzapas: array[1..2] of integer= (7,10);
typm:tkom20=(' MiG-21bis', ' MiG-29 ', '', ''
'', '', '', '', '', '', '', '', '', '', '', '',
czt:tkom20=(' TAK ',
' NIE ', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '');

czasna_z:tkom20=(' 5 ', ' 6 ', ' 7 ', ' 8 ', ' 9 ', ' 10'
'', '', '', '', '', '', '', '',

```



```
' 2 x R-73          +zb. ',
' 2 x R-60          ',
' 2 x R-60          +zb. ',
'FKM (R-27)        ',
'FKM (R-27)        +zb ');
```

War21:tkom30=

```
(' Bez zbiornikow  ',
' z 1 x 490 L      ',
' z 1 x 765 L      ',
' z 2 x 490 L      ',
' z 2 x 490 L+1 x 765 L ',
' z 3 x 490 L      ',',',',',',',',',
',',',',',',',',',',',',',',',',',
',',',',',',',',',',',',',',',',');
```

TTzb:array[1..17,1..4] of integer=

```
((9,0,6003,0),
(11,2,6308,150),
(11,2,6320,162),
(11,2,6183,90),
(12,3,6458,180),
(11,2,6342,184),
(12,3,6334,124),
(14,5,6638,270),
(12,3,6518,300),
(12,3,6542,324),
(16,7,6570,248),
(13,4,6522,274),
(13,4,6552,334),
(13,4,6564,346),
(12,3,6500,252),
(12,3,6530,312),
(12,3,6488,240));
```

Tzb:tkom30=

```
(' bez uzbrojenia  ',
' 2 x R-3s        ',
' 2 x R-3R        ',
' 2 x R-60M       ',
' 4 x R-60M       ');
```

```

' 2 x R-13M          ',
' 2 x UB-16          ',
' 6 x R-60M          ',
' 4 x R-3S           ',
' 4 x R-3R           ',
' 4 x UB-16          ',
' 2 x R-13M + 2 x R-60M ',
' 2 x R-13M + 2 x R-3S ',
' 2 x R-13M + 2 x R-3R ',
' 2 x R-3R + 2 x R-60M ',
' 2 x R-3R + 2 x R3S  ',
' 2 x R-3S +2 x R-60  ',
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '');

```

Tbzb:tkom30=

```

(' bez uzbrojenia    ',
' 2 x R-3s           ',
' 2 x R-3R           ',
' 2 x R-60M          ',
' 4 x R-60M          ',
' 2 x R-13M          ',
' 2 x UB-16          ',
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '');

```

Tzas:tkom30=

```

(' V maksymalnego zasięgu ',
' V nakazana          ',
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '' );

```

var

```

iii:real;
tel,ii:integer;
liczba:string[5];

woc,moo,p1,p2,tk,zb:integer;
dy,mst:real;
zlawartosc:integer;

```

Procedure dane\_1;

begin

```

wspy_bar:=0;
clearviewport;
rectangle(1,1,getmaxx,getmaxy);
skoki:=-1;
{1:skoki:=skoki+1;}
graphdefaults;
setviewport(2,2,getmaxx-1,getmaxy-1,clipon);
wspy_bar:=wspy_bar+10;
setcolor(0);
bar(8,5,620,20);
bar(8,25,620,210);
setlinestyle(1,0,3);
outtextxy(20,10,'          DANE          PRZYJETE
          DO          OBLICZEN');
outtextxy(10,30,'Typ myśliwca .....* '+zmienne[1]);
outtextxy(10,40,'Wariant uzbrojenia ..* '+zmienne[2]);
outtextxy(10,50,'Gęstość paliwa
.....* '+zmienne[3]);
outtextxy(10,60,'Czas pracy na ziemi
.....[min.]....* '+zmienne[4]);
outtextxy(10,70,'Zapasy nawigacyjny.
.....[% ]....* '+zmienne[5]);
outtextxy(10,80,'Średnia temp. w rejonie lotów
.[stop. C ]....* ');
outtextxy(10,90,'Średnia wysokość rejonu lotów n.p.m.
[m]....* ');
end;

```

```

Procedure ograniczenie(var flaga_ogr:integer);
var
i,j, k: integer;

begin
flaga_ogr:=0;
if Qp<350 then
begin
setcolor(15);
graphdefaults;

```

```

        rectangle(9,249,602,302);
        setviewport(10,250,600,300,clipon);
        clearviewport;

outtextxy(10,20,'Dla podanych warunkow nie
           ma mozliwosci dyzurowania');
outtextxy(300,30,'      Enter');
for k:=1 to 3 do begin

    for j:=1 to 25 do
    begin
        for i:=1 to 5 do
        begin
            sound(500+180*i);
            delay(3);
            nosound;
            sound(1500-180*i);
            delay(1);
            nosound;
        end;
        delay(5);
    end;
        delay(15);
            end;

            repeat
            r:=readkey;
            until r=#13;
            flaga_ogr:=1;
            graphdefaults;

            end;

end;

procedure minusik(a,b :integer; var ii:integer);

```

```

label
ss2, ss1, sss;

var

flaga, blad, i: integer;
p: char;
begin
    liczba:='';
    i:=1;
    flaga:=0;
    repeat
        begin
sss:      p:=readkey;
            if p=#8 then goto ss1;
            if p=#13 then goto ss2;
if (((p in ['0'..'9']) or (p = ('-'))
            or (p=('+')) or (p=('.')))) then
                goto ss1 else
                    begin
                        Sound(300);
                        Delay(100);
                        Nosound;
                        goto sss;
                    end;

ss1:      if p=#8 then
                begin
                    if i>1 then
                        begin
                            i:=i-1;
                            if i=2 then flaga:=0;
                            delete(liczba, i, 1);
                        end
                    else
                        begin
                            Sound(600);
                            Delay(100);

```

```

        Nosound;
    end;
end
else
    begin
        if ((i=1) and ((p in ['0'..'9'])
            or (p = ('-')) or (p=('+')))) then
            begin
                liczba:=liczba+p;
                val(liczba,i,blad);
                if ii>=999999 then
                    begin
                        Sound(600);
                        Delay(100);
                        Nosound;
                        i:=i-1;
                        delete(liczba,i,1);
                        goto sss;
                    end;

                    i:=i+1;
                end
            end
        else
            begin
                if flaga=0 then begin
                    if (((i=1) or (i=2)) and (p='.'))
                        then
                            begin
                                liczba:=liczba+p;
                                flaga:=1;
                            end;
                        end;
                    if (((p='.') and (flaga=1))and (i>2))
                        then
                            begin end
                        else
                            begin

```

```

        i:=i+1;

        end;
    if ((i>1) and (p in['0'..'9']))
        then
            begin
                liczba:=liczba+p;
                val(liczba,ii,blad);
                if ii>=999999 then
                    begin
                        Sound(600);
                        Delay(100);
                        Nosound;
                        i:=i-1;
                        delete(liczba,i,1);
                        goto sss;
                    end;
                end;
            else
                begin
                    if p=#8 then
                        begin
                            if i>1 then
                                begin
                                    i:=i-1;
                                    delete(liczba,i,1);
                                end
                            else
                                begin
                                    end;
                                end;
                            end;
                end;
            end;
        end;
    if (p<>#13)or((p=#13)and(i=1)) then
        begin
            Sound(400);
            Delay(100);
            Nosound;

```

```

        end;
    end;
    SetFillStyle(0,0);
    bar(a,b,a+60,b+10);
    OutTextXY(a,b,liczba);
    SetFillStyle(1,1);
end;
until (i>7) or ((p=#13)and(i>1));
ss2:   val(liczba,ii,blad);

end;

procedure min_b(a,b :integer; var ii:integer);
label
ss2,ss1,sss;

var

flaga,blad,i:integer;
p:char;
begin
    setcolor(0);
    bar(a,b-2,a+60,b+12);
    line(a,b+10,a+50,b+10);
    liczba:='';
    i:=1;
    flaga:=0;
    repeat
        begin
ss2:   p:=readkey;
            if p=#8 then goto ss1;
            if p=#13 then goto ss2;
            if ( ((p in ['0'..'9']) or (p = ('-'))
                or (p=('+')))) then

                                goto ss1 else

```

```

begin
    Sound(300);
    Delay(100);
    Nosound;
    goto sss;
end;

ss1:  if p=#8 then
        begin
            if i>1 then
                begin
                    i:=i-1;
                    if i=2 then flaga:=0;
                    delete(liczba,i,1);
                    Sound(600);
                    Delay(100);
                    Nosound;
                end;
            end
        else
            begin

if ((i=1) and ((p in ['0'..'9'])
or (p = ('-')) or (p=('+')))) then
                begin
                    liczba:=liczba+p;
                    val(liczba,ii,blad);
                    if ii>=999999 then
                        begin
                            Sound(600);
                            Delay(100);
                            Nosound;
                            i:=i-1;
                            delete(liczba,i,1);
                            goto sss;
                        end;

                    i:=i+1;
                end
            end
        end

```

```

else
  begin
    if flaga=0 then begin

if (((i=1) or (i=2)) and (p='.')) then
  begin
    liczba:=liczba+p;
    flaga:=1;
  end;

  end;
if (((p='.') and (flaga=1))and (i>2)) then
  begin end
  else
  begin
    i:=i+1;

  end;
if ((i>1) and (p in['0'..'9'])) then
  begin
    liczba:=liczba+p;
    val(liczba, ii, blad);
    if ii>=999999 then
      begin
        Sound(600);
        Delay(100);
        Nosound;
        i:=i-1;
        delete(liczba, i, 1);
        goto sss;
      end;
    end
  else
  begin
    if p=#8 then
      begin
        if i>1 then
          begin
            i:=i-1;
            delete(liczba, i, 1);

```

```

end;
end;
end;

end;
setcolor(0);
bar(a, b-2, a+60, b+12);
line(a, b+10, a+50, b+10);
OutTextXY(a, b, liczba);
end;
until (i>7) or ((p=#13)and(i>1));
ss2: val(liczba, ii, blad);
setcolor(15);

end;

procedure oknodol;
begin
setcolor(15);
graphdefaults;
rectangle(9, 249, 602, 302);
setviewport(10, 250, 600, 300, clipon);
clearviewport;
end;

procedure scroll30(il, px, py, kx, ky: integer;
obiekt:tkom30; var k: integer);

var
i, m, n, z : integer;
begin

SetFillStyle(0, 0);
k:=1;
m:=1;
if il<18 then n:=il
else n:=18;
repeat
for i:=m to n do
begin

```

```

    bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
if i=k then
    begin
        setcolor(0);
        SetFillStyle(1,1);
        setcolor(0);
        bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
        SetFillStyle(0,0);
    outtextxy(px+10, py+4+(i-m)*15, obiekt[i]);
        SetColor(1);
    end
else
    outtextXY(px+4, py+4+(i-m)*15, obiekt[i]);
end;
OuttextXY(10, 333, 'WYBIERZ WARIANT I NACISNIJ "Enter" ');
repeat
    r:=readkey;
    if r=#0 then
        begin
            r:=readkey;
            z:=0;
            case r of
                'I' : begin
                    k:=1;
                    m:=1;
                    if il<18 then n:=il
                        else n:=18;
                    z:=1;
                end;
                'Q' : begin
                    k:=il;
                    if il<18 then m:=1
                        else m:=il-17;
                    n:=il;
                    z:=1;
                end;
                'H' : begin
                    k:=k-1;
                    if k=0 then

```

```

begin
  k:=1;
  Sound(800);
  Delay(300);
  Nosound;
end
else
begin
  z:=1;
  if k<m then begin
    m:=m-1;
    n:=n-1;
  end
  else
  begin
    z:=0;
    SetFillStyle(1,1);
    bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
    SetFillStyle(0,0);
    SetColor(0);
    outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
    SetColor(1);
    bar(px,py+(k-m+1)*15,kx,ky+(k-m+1)*15);
    outtextXY(px+4,py+4+(k-m+1)*15,obietk[k+1]);
  end;
end;
end;
'P' : begin
  k:=k+1;
  if k>il then
  begin
    k:=il;
    Sound(800);
    Delay(300);
    Nosound;
  end
else
begin
  z:=1;

```

```

        if k>n then begin
            n:=n+1;
            m:=m+1;
            end
        else
            begin
                z:=0;
                SetFillStyle(1,1);
                bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
                SetFillStyle(0,0);
                SetColor(0);
                outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
                SetColor(1);
                bar(px,py+(k-m-1)*15,kx,ky+(k-m-1)*15);
                outtextXY(px+4,py+4+(k-m-1)*15,obietk[k-1]);
                end;
            end;
        end;
    end
end
else
    if r<>#13 then
        begin
            Sound(800);
            Delay(300);
            Nosound;
        end;
    until (r=#13)or(z=1)
until r=#13;
SetFillStyle(1,1);

end;

procedure scroll20(il,px,py,kx,ky: integer;
                 obietk:tkom20; var k: integer);
var
i,m,n,z : integer;
begin

```

```

        SetFillStyle(0,0);
        k:=1;
        m:=1;
if il<12{21} then n:=il
        else n:=12{21}; {bylo 15}
        repeat
for i:=m to n do
        begin
            bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
            if i=k then
                begin
                    setcolor(0);
                    SetFillStyle(1,1);
                    setcolor(0);
                    bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
                    SetFillStyle(0,0);
                outtextxy(px+10,py+4+(i-m)*15,obiekt[i]);
                    SetColor(1);
                end
            else
                outtextXY(px+4,py+4+(i-m)*15,obiekt[i]);
            end;
OuttextXY(10,333,'WYBIERZ WARIANT I NACISNIJ "Enter" ');
        repeat
            r:=readkey;
            if r=#0 then
                begin
                    r:=readkey;
                    z:=0;
                    case r of
                        'I' : begin
                            k:=1;
                            m:=1;
                            if il<12 then n:=il
                                else n:=12;{15;}
                            z:=1;
                        end;
                        'Q' : begin
                            k:=il;

```

```

        if il<12{15} then m:=1
            else m:=il-11; {14}
        n:=il;
        z:=1;
    end;
'H' : begin
    k:=k-1;
    if k=0 then
        begin
            k:=1;
            Sound(800);
            Delay(300);
            Nosound;
        end
    else
        begin
            z:=1;
            if k<m then begin
                m:=m-1;
                n:=n-1;
            end
            else
                begin
                    z:=0;
                    SetFillStyle(1,1);
                    bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
                    SetFillStyle(0,0);
                    SetColor(0);
                    outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
                    SetColor(1);
                    bar(px,py+(k-m+1)*15,kx,ky+(k-m+1)*15);
                    outtextXY(px+4,py+4+(k-m+1)*15,obietk[k+1]);
                end;
            end;
        end;
    end;
'P' : begin
    k:=k+1;
    if k>il then
        begin

```

```

        k:=il;
        Sound(800);
        Delay(300);
        Nosound;
    end
else
    begin
        z:=1;
        if k>n then begin
            n:=n+1;
            m:=m+1;
        end
        else
            begin
                z:=0;
                SetFillStyle(1,1);
                bar(px,py+(k-m)*15,kx,ky+(k-m)*15);

                SetFillStyle(0,0);
                SetColor(0);
                outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
                SetColor(1);
                bar(px,py+(k-m-1)*15,kx,ky+(k-m-1)*15);
                outtextXY(px+4,py+4+(k-m-1)*15,obietk[k-1]);

                end;
            end;
        end;
    end
end
else
    if r<>#13 then
        begin
            Sound(800);
            Delay(300);
            Nosound;
        end;
    until (r=#13)or(z=1)
until r=#13;
SetFillStyle(1,1);

```

```

end;
procedure scroll7(il,px,py,kx,ky: integer;
  obiekt:tkom30; var k: integer);
var
i,m,n,z : integer;
begin

      SetFillStyle(0,0);
      k:=1;
      m:=1;
  if il<6 then n:=il
      else n:=6; {bylo 15}
      repeat
  for i:=m to n do
  begin
      bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
  if i=k then
      begin
          setcolor(0);
          SetFillStyle(1,1);
          setcolor(0);
          bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
          SetFillStyle(0,0);
          outtextxy(px+10,py+4+(i-m)*15,obiekt[i]);
          SetColor(1);
      end
      else
          outtextXY(px+4,py+4+(i-m)*15,obiekt[i]);
  end;
OuttextXY(10,333,'WYBIERZ WARIANT I NACISNIJ "Enter" ');
  repeat
      r:=readkey;
  if r=#0 then
      begin
          r:=readkey;
          z:=0;
          case r of
              'I' : begin

```

```

        k:=1;
        m:=1;
        if il<9 then n:=il
            else n:=9;{15;}
        z:=1;
    end;
'Q' : begin
    k:=il;
    if il<9{15} then m:=1
        else m:=il-8; {14}
    n:=il;
    z:=1;
end;
'H' : begin
    k:=k-1;
    if k=0 then
        begin
            k:=1;
            Sound(800);
            Delay(300);
            Nosound;
        end
    else
        begin
            z:=1;
            if k<m then begin
                m:=m-1;
                n:=n-1;
            end
            else
                begin
                    z:=0;
                SetFillStyle(1,1);
                bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
                SetFillStyle(0,0);
                SetColor(0);
                outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
                SetColor(1);
                bar(px,py+(k-m+1)*15,kx,ky+(k-m+1)*15);

```

```

outtextXY(px+4, py+4+(k-m+1)*15, obiekt[k+1]);
end;
end;
end;
'P' : begin
k:=k+1;
if k>il then
begin
k:=il;
Sound(800);
Delay(300);
Nosound;
end
else
begin
z:=1;
if k>n then begin
n:=n+1;
m:=m+1;
end
else
begin
z:=0;
SetFillStyle(1,1);
bar(px, py+(k-m)*15, kx, ky+(k-m)*15);

SetFillStyle(0,0);
SetColor(0);
outtextXY(px+4, py+4+(k-m)*15, obiekt[k]);
SetColor(1);
bar(px, py+(k-m-1)*15, kx, ky+(k-m-1)*15);
outtextXY(px+4, py+4+(k-m-1)*15, obiekt[k-1]);
end;
end;
end;
end
else
if r<>#13 then

```

```

begin
    Sound(800);
    Delay(300);
    Nosound;
end;
until (r=#13)or(z=1)
until r=#13;
SetFillStyle(1,1);

end;
procedure okno3;
begin
setviewport(7,10,289,210,clipon);
end;
procedure daneztabeli21;
begin
Woc:=TTzb[iltzb,1];
Wocu:=TTzb[iltzb,2];
Mst:=TTzb[iltzb,3];
Mstu:=TTzb[iltzb,4];
end;

procedure daneztabeli29(k:integer ;var mst,dy:real;
var woc,moo,p1,p2,tk,zb,mr,wocr:integer);
begin
mst:=0;
woc:=0;
mst:=Tmst[k,1];
woc:=trunc(Tmst[k,2]);
moo:=trunc(Tmst[k,3]);
p1:=trunc(Tmst[k,4]);
p2:=trunc(Tmst[k,5]);
dy:=trunc(Tmst[k,6]);
tk:=trunc(Tmst[k,7]);
zb:=trunc(Tmst[k,8]);
mr:=trunc(Tmst[k,9]);
wocr:=trunc(Tmst[k,10]);
end;

```

```

begin
driver:=5;
start:initgraph(driver,mode,'');
flaga_ogr:=0;
clearviewport;
pocz1(iltabpocz);
rectangle(1,1,getmaxx,getmaxy);
graphdefaults;
outtextxy(40,30,' Wybierz');
outtextxy(40,40,'TYP myśliwca');
rectangle(40,60,140,96);
scroll20(2,42,62,138,79,typm,iltypm);
    zmienne[1]:=typm[iltypm];
    case iltypm of
        1:begin {MiG-21bis}
            graphdefaults;
            outtextxy(200,20,'WARIANT UZBROJENIA');
            outtextxy(170,30,'+ komplet amunicji do działka');
            rectangle(150,40,402,320);
            setviewport(151,41,401,319,clipon);
            rectangle(10,10,201,110);
            scroll30(6,11,11,200,28,war21,iluz);
            case iluz of
                1:zb:=0;
                2..6:zb:=1;
            end;
            clearviewport;
            graphdefaults;

            case iluz of
                0,1,2:begin
                    scroll130(17,152,42,400,59,Tzb,iltzb);
                    zmienne[2]:=Tzb[iltzb];
                    end;
                3..5:begin
                    scroll130(7,152,42,400,59,Tbzb,iltzb);
                    zmienne[2]:=Tbzb[iltzb];
                    end;
            end;
        end;
end;

```

```

daneztabeli21;
end;

2:begin {MiG-29}
graphdefaults;
outtextxy(200,20,'WARIANT UZBROJENIA');
outtextxy(170,30,'+ komplet amunicji do dzialka');
rectangle(150,40,402,320);
scroll30(30,152,42,400,59,wariant,ilwariant);
zmienne[2]:=wariant[ilwariant];
k:=ilwariant;
daneztabeli29(k,mst,dy,woc,moo,p1,p2,tk,zb,mr,wocr);
end;
end; {od case iltypm}

graphdefaults;
outtextxy(420,20,'GESTOSC');
outtextxy(420,30,'PALIWA');
rectangle(420,40,472,200);
scroll30(10,422,42,470,59,gestosc,ilgestosc);
zmienne[3]:=gestosc[ilgestosc];
graphdefaults;
outtextxy(500,20,'Czas pracy');
outtextxy(500,30,'na ziemi [ min. ]');
rectangle(502,40,580,137);
scroll20(6,504,42,578,59,czasna_z,ilczasna_z);
zmienne[4]:=czasna_z[ilczasna_z];
graphdefaults;

outtextxy(450,220,'Zapas nawigacyjny');
rectangle(490,240,540,275);
scroll20(2,492,242,538,259,zapas,ilzapas);
zmienne[5]:=zapas[ilzapas];
zn:=Tzapas[ilzapas]/100;

dane_1;
poczatek:

```

```

outtextxy(10,100,'Paliwo walki powietrznej.
.....[kg].....* ');
outtextxy(10,110,'Wysokosc dyzuruwania.
.....[m].....* ');

outtextxy(10,120,'Predkosc dyzuruwania.
.....[km/h]....* ');
outtextxy(10,130,'Odleglosc strefy dyzuruwania.
.....[km].....* ');
outtextxy(10,140,'Wysokosc dolotu do strefy.
.....[m].....* ');
outtextxy(10,150,'Predkosc dolotu do strefy.
.....[km/h]....* ');
if iltabpocz=1 then
outtextxy(10,160,'Odleglosc do lotniska ladowania.
.....[km ].....* ');
outtextxy(10,170,'Wysokosc powrotu.
.....[m].....* ');
outtextxy(10,180,'Predkosc powrotu.
.....[km/h]....* ');

oknodol;
if flaga_ogr=1 then goto W3;
w1:
outtextxy(20,20,'Srednia temperatura w rejonie lotow..');

min_b(400,20,TEL);
if ((tel<0) or (tel>30)) then
begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
outtextxy(20,20,'(Mozliwy zakres (0..+30) Nacisnij ENTER');
repeat
r:=readkey;
until r=#13;
clearviewport;setcolor(15);goto w1;
end

```

```

else
    begin
        graphdefaults;
        setcolor(0);
        str(tel:4,stel);
        outtextxy(400,80,''+stel);
        zmienne[6]:=stel;
        end;

    oknodol;

W2:
outtextxy(20,20,'Srednia wysokosc rejonu lotow
    [metry]. ');
    min_b(400,20,H1);
    if ((H1<0) or (H1>3000)) then
        begin
            clearviewport;
            setcolor(0);
            bar(1,1,600,50);
outtextxy(20,20,'(Mozliwy zakres (0..3000)metrow
    Nacisnij ENTER');

            repeat
                r:=readkey;
                until r=#13;
                clearviewport;setcolor(15);
                goto w2;
            end
        else
            begin
                graphdefaults;
                setcolor(0);
                str(H1:4,stel);
                outtextxy(400,90,''+stel);
                zmienne[7]:=stel;
                end;

    oknodol;

W3:

```

```

outtextxy(20,20,'Paliwo walki powietrznej [kg] ..');
min_b(400,20,Qw);
if ((Qw<0) or (Qw>1500)) then
    begin
        clearviewport;
        setcolor(0);
        bar(1,1,600,50);
outtextxy(20,20,'(Mozliwy zakres (0..1500)kilogramow
Nacisnij ENTER');
        repeat
            r:=readkey;
            until r=#13;
        clearviewport;setcolor(15);
        goto w3;
        end
    else
        begin
            graphdefaults;
            setcolor(0);
            str(Qw:4,stel);
            outtextxy(400,100,''+stel);
            zmienne[8]:=stel;
            end;

case iltypm of
1:begin
    case zb of
    0: Begin
        moo:=2750; mz:=0; Wocz:=0;
        end;
    1:Begin
        moo:=3240; mz:=70; Wocz:=2;
        end;
    2:Begin
        moo:=3515; mz:=85; Wocz:=2;
        end;
    3:Begin
        moo:=3730; mz:=92; Wocz:=3;
        end;

```

```

4: Begin
    moo:=4495;mz:=177;Wocz:=5;
    end;
5: Begin
    moo:=4220;mz:=162;Wocz:=5;
    end;

end; {case od 1;}
    Qz:=10; Tlad:=300; Qlad:=200;
    Mst:=Mst+112+mz;
    Woc:=Woc+Wocz;
end;
2: begin
    if zb=0 then moo:=5700 else moo:=4200;
    Qz:=20; Tlad:=315; Qlad:=160;
    end;
end; {od calego case}
Mstwe:=Mst;
gamma:=tgest[ilgestosc];
Qp:=0.93*moo*gamma-Tczasna_z[ilczasna_z]*Qz;
    {z zapasem tech}
Mst:=Mst+moo*gamma-Tczasna_z[ilczasna_z]*Qz;
w7: oknodol;
outtextxy(20,20,'Wysokosc dyzurowania [m] ..');
    min_b(400,20,H22);
if ((H22<200) or (H22>9000-HL)) then
    begin
        clearviewport;
        setcolor(0);
        bar(1,1,600,50);
        str(9000-Hl:5,stel);
outtextxy(20,20,'(Mozliwy zakres
( 200 m. ..'+stel+' ) Nacisnij ENTER');
        repeat
            r:=readkey;
            until r=#13;
        clearviewport;setcolor(15);goto w7;
        end

```

```

else
begin
graphdefaults;
setcolor(0);
str(H22:4,stel);
outtextxy(400,110,''+stel);
zmienne[12]:=stel;
end;

{#####}
H1:=H22+H1; TE:=TEL-15+0.0065*H1; if TE<0 then TE:=0;
TEM2:= TEL-0.0065*(H1+H22)+273; TEM:=TEM2;
mst:=mst-0.5*Qp;
case iltypm of
1: maksymalnyzasieg21(Woc,H1,TEM,Vmkk);
2: maksymalnyzasieg29(mst,H1,TEM,Vmkk);
end;
mst:=mst+0.5*Qp;
V2mz:=trunc(Vmkk);
Vmax:=0.9*72.2*sqrt(TEM2);

W8: oknodol;
outtextxy(20,20,'Wybierz rodzaj predkosci dyzurowania');
rectangle(389,9,587,44);
scroll130(2,390,10,585,27,Tzas,ilTzas);
clearviewport;
if ilTzas=2 then
begin
outtextxy(20,20,'Predkosc dyzurowania [km/h] ..');
min_b(400,20,V2);
if ((V2<V2mz) or (V2>Vmax)) then
begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
str(V2mz:5,stel);
str(Vmax:5:0,sVmax);
outtextxy(20,20,'(Mozliwy zakres ('+stel+'..'

```

```

+svmax+ ' km/h)'+ ' Nacisnij ENTER');
    repeat
        r:=readkey;
        until r=#13;
    clearviewport;setcolor(15);goto w8;
    end
    else
        begin
            graphdefaults;
            setcolor(0);
            str(V2:4,stel);
            outtextxy(400,120,''+stel+' ( V nakazana )');
            zmienne[13]:=stel;
            end;
        end
    else
        begin
            graphdefaults;
            setcolor(0);
            V2:=V2mz;
            str(V2:4,stel);
            outtextxy(400,120,''+stel+' ( V maks. zasiegu )');
            zmienne[13]:=stel;
            end;
        end;
w4:   oknodol;
outtextxy(20,20,'Odleglosc strefy dyzowania [km]..');
    min_b(400,20,S1);
    if (S1<=0) then
        begin
            clearviewport;
            setcolor(0);
            bar(1,1,600,50);
            outtextxy(20,20,'(Wprowadzona dana jest poza zakresem
            Nacisnij ENTER)');
            repeat
                r:=readkey;

```

```

        until r=#13;
clearviewport;setcolor(15);goto w4;
        end;
if (S1>11000) then
        begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
outtextxy(20,20,'(Za duza odleglosc Nacisnij ENTER)');
repeat
r:=readkey;
until r=#13;
clearviewport;setcolor(15);goto w4;
end;
graphdefaults;
setcolor(0);
str(S1:4,stel);
outtextxy(400,130,''+stel);
zmienne[9]:=stel;

w5: oknodol;
outtextxy(20,20,'Wysokosc dolotu do strefy [m] ');
min_b(400,20,H11);
if ((H11<H22) or (H11>9000-H1)) then
        begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
str(H22:5,stel);
str(9000-H1:5,stel1);
outtextxy(20,20,'(Mozliwy zakres ('+stel+'..' +stel1+'m)
Nacisnij ENTER)');
repeat
r:=readkey;
until r=#13;
clearviewport;setcolor(15);goto w5;
end
else
        begin

```

```

graphdefaults;
setcolor(0);
str(H11:4,stel);
outtextxy(400,140,''+stel);
zmienne[10]:=stel;
end;

H1:=H11+H1; TE:=TEL-15+0.0065*H1; if TE<0 then TE:=0;
TEM1:= TEL-0.0065*(H1+H11)+273; TEM:=TEM1;
case iltypm of
1: maksymalnyzasieg21(Woc,H1,TEM,Vmkk);
2: maksymalnyzasieg29(mst,H1,TEM,Vmkk);
end;
V1mz:=trunc(Vmkk); {rzeczywista}
Vmax:=0.9*72.2*sqrt(TEM1);
w6: oknodol;
outtextxy(20,20,'Wybierz rodzaj predkosci dolotu do strefy');
rectangle(389,9,587,44);
scroll30(2,390,10,585,27,Tzas,ilTzas);
clearviewport;
if ilTzas=2 then
begin
outtextxy(20,20,'Nakazana predkosc
dolotu do strefy [km/h] ..');
min_b(400,20,V1);
if ((V1<V1mz) or (V1>Vmax)) then
begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
str(V1mz:5,stel);
str(Vmax:5:0,sVmax);
outtextxy(20,20,'(Mozliwy zakres ('+stel+'..'sVmax+'
km/h)'+ Nacisnij ENTER');
repeat
r:=readkey;
until r=#13;
clearviewport;setcolor(15);goto w6;
end

```

```

else
begin
graphdefaults;
setcolor(0);
str(V1:4,stel);
outtextxy(400,150,''+stel+' ( V nakazana )');
zmienne[11]:=stel;
end;

end
else {V max zasięgu}
begin
graphdefaults;
setcolor(0);
V1:=V1mz;
str(V1:4,stel);
outtextxy(400,150,''+stel+' ( V maks. zasięgu )');
zmienne[11]:=stel;

end;
if iltabpocz=1 then
begin
w9: oknodol;

outtextxy(20,20,'Odleglosc do lotniska
ladowania [km] ');
min_b(400,20,S3);
if (S3<=0) then
begin
clearviewport;
setcolor(0);
bar(1,1,600,50);
outtextxy(20,20,'(Mozliwy zakres (>0)
Nacisnij ENTER');

repeat
r:=readkey;
until r=#13;
clearviewport;setcolor(15);goto w9;
end

```

```

        else
            begin
                graphdefaults;
                setcolor(0);
                str(S3:4,stel);
                outtextxy(400,160,''+stel);
                zmienne[14]:=stel;
            end;

        end

        else
            S3:=S1;

w10:   oknodol;
        outtextxy(20,20,'Wysokosc powrotu [m] ..');
        min_b(400,20,H33);
        if ((H33<H22) or (H33>9000-H1)) then
            begin
                clearviewport;
                setcolor(0);
                bar(1,1,600,50);
                str(H22:5,stel);
                str(H22:5,stel);
                str(9000-H1:5,stel1);
                outtextxy(20,20,'(Mozliwy zakres ('+stel+'
                ..'+stel1+'m)  Nacisnij ENTER');

                repeat
                    r:=readkey;
                until r=#13;

                clearviewport;setcolor(15);goto w10;
            end
        else
            begin
                graphdefaults;
                setcolor(0);
                str(H33:4,stel);
                outtextxy(400,170,''+stel);
                zmienne[15]:=stel;
            end;
        end;
    end;
end;

```

```

                                end;

{#####}
H1:=H33+H1; TE:=TEL-15+0.0065*H1; if TE<0 then TE:=0;
TEM3:= TEL-0.0065*(H1+H33)+273; TEM:=TEM3;
                                y:=mst;
                                mst:=mstwe+600;
                                case iltypm of
1: maksymalnyzasieg21(Woc,H1,TEM,Vmkk);
2: maksymalnyzasieg29(mst,H1,TEM,Vmkk);
                                end;
                                mst:=y;
                                V3mz:=trunc(Vmkk);
                                Vmax:=0.9*72.2*sqrt(TEM3);

w11: oknodol;

                                outtextxy(20,20,'Wybierz rodzaj predkosci powrotu');
                                rectangle(389,9,587,44);
                                scroll130(2,390,10,585,27,Tzas,ilTzas);
                                clearviewport;
                                if ilTzas=2 then
                                begin
outtextxy(20,20,'Predkosc powrotu      [km/h]      ..');
                                min_b(400,20,V3);
                                if ((V3<V3mz) or (V3>Vmax)) then
                                begin
                                        clearviewport;
                                        setcolor(0);
                                        bar(1,1,600,50);
                                        str(V3mz:5,stel);
                                        str(Vmax:5:0,sVmax);
outtextxy(20,20,'(Mozliwy zakres ('+stel+'
..' +sVmax+' km/h)'+ ' Nacisnij ENTER');

                                        repeat
                                        r:=readkey;
                                        until r=#13;

                                clearviewport;setcolor(15);goto w11;
                                end

```

```

else
    begin
        graphdefaults;
        setcolor(0);
        str(V3:4,stel);
        outtextxy(400,180,''+stel+' ( V nakazana )');
        zmienne[16]:=stel;
        end;

end
else
    begin
        graphdefaults;
        setcolor(0);
        V3:=V3mz;
        str(V3:4,stel);
        outtextxy(400,180,''+stel+' ( V maks. zasiegu )');
        zmienne[16]:=stel;

end;

11:
{*****koniec wprowadzania danych *****}

H11:=H11+H1; H22:=H22+H1; H33:=H33+H1;
V1M:=V1/(72.2*sqrt(TEM1));
V1P:=ln(V1)-0.5*ln(TEM1)+2.831+2.628*ln(1-0.0000226*H11);
V1P:=exp(V1P);

V2M:=V2/(72.2*sqrt(TEM2));
V2P:=ln(V2)-0.5*ln(TEM2)+2.831+2.628*ln(1-0.0000226*H22);
V2P:=exp(V2P);

V3M:=V3/(72.2*sqrt(TEM3));
V3P:=ln(V3)-0.5*ln(TEM3)+2.831+2.628*ln(1-0.0000226*H33);
V3P:=exp(V3P);

{***** start-bez czasu *****}

```

```

        case iltypm of
1: y:=2*(Woc-9)+140;
2: y:=0.25*(Woc-230)+0.01*(Mst-13000)+1.2*TE+140;
        end;
        Qp:=Qp-y; Mst:=Mst-y;

{***** odcinek 1 *****}

        if H22<1000 then
                begin
                        if H11>1000 then
                                begin
                                        Hkk:=1000;
                                        Hp:=H11;
                                        case iltypm of
1: znizanie21b(Hkk, Hp, Tzn, Qzn, Szn);
2: znizanie29(Hkk, Hp, Tzn, Qzn, Szn);
                                        end;

                                        Szn:=Szn+0.02*(1000-H22);
                                        Tzn:=Tzn+3600*Szn/V1;
                                        Qzn:=Qzn+Tzn*35/60;
                                        end
                                        else
                                                begin
                                                        Szn:=0.02*(H11-H22);
                                                        Tzn:=3600*Szn/V1;
                                                        Qzn:=Tzn*35/60;
                                                        end;
                                                end
                        end
                        else
                                begin
                                        Hkk:=H22;
                                        Hp:=H11;
                                        case iltypm of
1: znizanie21b(Hkk, Hp, Tzn, Qzn, Szn);
2: znizanie29(Hkk, Hp, Tzn, Qzn, Szn);
                                        end;
                                end;
                end;

```

```

if H11>200 then
    begin
        Hp:=200+H1; Hkk:=H11;
        if Hkk=Hp then begin
            Swzn:=0; Twzn:=0; Qwzn:=0;
            end
        else
            begin
                case ilitypm of
                    1:wznoszenie21b(Hkk, Hp, woc, mst, Qp, Swzn, Qwzn, Twzn);
                    2:wznoszenie29(gamma, Hp, Hkk, TE,
                        moo, Woc, Qp, mst,
                        Swzn, Twzn, Qwzn);
                end;
            end;
            H1:=Hkk; Qp:=Qp-zn*Qwzn;
            end
        else
            begin
                H1:=200+H1; Swzn:=0; Qwzn:=0; Twzn:=0;
                end;
                Vmkk:=V1M; Vp:=V1P;
                case ilitypm of
                    1:oblicz_ck21b(H1, Vp, mst, Woc, ck);
                    2:oblicz_ck29(Mst, H1, Vmkk, Woc, Ck);
                end;
                y:=S1-Swzn-Szn;
                if y<0 then y:=0;
                Mst:=Mst-y*ck-Qzn;
                Qp:=Qp-(y*ck+Qzn)*(1-zn); Ta1:=Tzn+Twzn+3600*y/V1;
                Mstp:=Mst;
                Qpp:=Qp;
                {***** odcinek 3 *****}
                ograniczenie(flaga_ogr);
                if flaga_ogr=1 then
                    begin
                        dane_1;
                    end;
                outtextxy(10, 80, '          '+zmienne[6]);
            end;
        end;
    end;

```

```

outtextxy(10,90,'
+zmienne[7]);
goto poczatek;
end;
Mst:=Mstwe+355;
Qp:=355;
if H33>(2000+H1) then
begin
Hkk:=2000+H1; Hp:=H33;
case iltypm of
1:znizanie21b(Hkk, Hp, Tzn, Qzn, Szn);
2:znizanie29(Hkk, Hp, Tzn, Qzn, Szn);
end;

mst:=mst+Qzn+Qlad;
Qp:=Qp+(Qzn+Qlad)*(1+zn);
Szn:=Szn+40;
Tzn:=Tzn+Tlad;
end
else
begin
case iltypm of
1:Qzn:=0.1*H33;
2:Qzn:=0.08*H33;
end;
Szn:=0.02*H33; Tzn:=7.2*Szn;
Qp:=Qp+(1+zn)*Qzn;
mst:=mst+Qzn;
end;
H1:=H33; Vmkk:=V3M; Vp:=V3P;
case iltypm of
1:oblicz_ck21b(H1, Vp, mst, Woc, ck);
2:oblicz_ck29(Mst, H1, Vmkk, Woc, Ck);
end;

y:=S3-Szn;
if y<0 then y:=0;
mst:=mst+y*ck/2;

```

```

        case ilytpm of
1: oblicz_ck21b(H1, Vp, mst, Woc, ck);
2: oblicz_ck29(Mst, H1, Vmkk, Woc, Ck);
        end;
        mst:=mst+y*ck;
        Hkk:=H33; Hp:=H22;
        if Hkk=Hp then begin
            Swzn:=0; Twzn:=0; Qwzn:=0;
            end
            else
            begin
                case ilytpm of
1: wznoszenie21b(Hkk, Hp, woc, mst, Qp, Swzn, Qwzn, Twzn);
2: wznoszenie29(gamma, Hp, Hkk, TE,
                moo, Woc, Qp, mst,
                Swzn, Twzn, Qwzn);
                end;
            end;
        end;
        y:=y-Swzn; if y<0 Then y:=0;

        mst:=mst+2*Qwzn -Swzn*ck;
        Qp:=Qp+(2*Qwzn+y*ck)*(1+zn);
        Ta3:=Tzn+Twzn+y*3600/V3;

        mst:=(Mstp+mst)/2;
        Qp:=Qpp-Qp-Qw;
        ograniczenie(flaga_ogr);
        if flaga_ogr=1 then
            begin
                dane_1;
                outtextxy(10, 80, '      '+zmienne[6]);
                outtextxy(10, 90, '      '+zmienne[7]);
                goto poczatek;
            end;
        H1:=H22; Vmkk:=V2M; Vp:=V2P;
        case ilytpm of
1: oblicz_ck21b(H1, Vp, mst, Woc, ck);
2: oblicz_ck29(Mst, H1, Vmkk, Woc, ck);

```

```

end;

y:=ck;

case ilypm of
1: manewr21b(H1, Vp, Vmkk, V3, ck, Tm);
2: manewr29(V3, Mst, H1, Vmkk, Woc, ck, Tm);
end;

ck:=(ck*Tm+240*y)/(240+Tm);
Tdyz1:=3600*Qp/(ck*V2);

Td:=(Ta1+Ta3)/30+0.5*Tczasna_z[ilczasna_z];

if ilypm=2 then begin
if zb=0 then zb:=1 else zb:=0;
end;

case iltabpocz of
1: begin
oknodol;
Tdyz1:=trunc(Tdyz1/60);
str(Tdyz1:5:0, sTdyz1);
outtextxy(20,20,'Czas dyzowania pojedynczego
samolotu'+typm[ilypm]+' =' +sTdyz1+' min');
repeat
r:=readkey;
until r=#13;

graphdefaults;
setviewport(3,230,getmaxx,getmaxy-4,clipon);
clearviewport;
outtextxy(10,40,'Czy powtarzasz obliczenia ?');
rectangle(48,58,152,94);
scroll20(2,50,60,150,77,czt,kut);
if kut=1 then goto start
else goto kkkk;

end;
2,3:begin
Tdyz1:=Tdyz1/60;
stary_program(k,iltabpocz,ilypm,zb,tdyz1,td);

{stary program dyzur}
end;

```

```
end;  
if flag=1 then goto start;  
closegraph;  
kkkk:  
closegraph;  
end.
```

```

unit stary;

interface
uses
pocz, graph, crt, printer, typy_n;
type
tkom150=array[1..20]of string[40];
tkomwydr=array[1..50] of string[30];
var
tabscr:tkomwydr;
T01, ppx, ppy, drukuj, T00, TT, ob,
wybor, kut, ilek, ile, ii: integer;
ssss, sstel, st0, std, stt, sn, sg, stx, stp, st,
ss, slp, liczba: string[5];
ilczyzm: integer;
r: char;
s: integer;
juzbylo, lp, czas, zb, taknie, a, iy, iio, io,
iii, il, b, T1, T2, T3, N: integer;
n1, tx1, T0, tp, t, TX, ng: real;
tdyz, tprzerwy: array[1..50]of real;
const
tab1: array[1..3, 1..6] of integer=
((25, 20, 25, 20, 25, 20),
(35, 30, 35, 30, 35, 30),
(90, 60, 90, 60, 90, 60));

tabstr: array[1..3, 1..6] of string[5]=
((' 25', ' 20', ' 25', ' 20', ' 25', ' 20'),
(' 35', ' 30', ' 35', ' 30', ' 35', ' 30'),
(' 90', ' 60', ' 90', ' 60', ' 90', ' 60'));
czyzm:tkom150=(' Tak', ' Nie', '', '', ''
, '', '', '', '', '', '', '', '', '', '', '',
'', '');
typsam:tkom150=
(' Mig-21', ' Mig-23', ' Mig-29', '', '', '', ''
, '', '', '', '', '', '', '', '', '', '', '');
ugrup:tkom150=
(' PARA', ' KLUCZ', ' ESKADRA', '', '', '', ''
, '', '', '', '', '', '', '', '', '', '', '');

```





```

Procedure a_1;
begin
    clearviewport;
    graphdefaults;
    setcolor(0);
    bar(ppx, 285, ppx+70, 300);
    str(T1:3, ssss);
    outtextxy(ppx+20, 290, +sss);
    setcolor(15);
end;

Procedure Tab_odtwarz;
begin
    outtextxy(100, 170, ' CZASY ODTWARZANIA GOTOWOSCI BOJOWEJ' );
    outtextxy(100, 200, '      Mig-21bis      Mig-29      ');
    outtextxy(100, 210, ' Ze zb.   Bez zb      Ze zb.   Bez zb. ');
    rectangle(100, 190, 460, 220);
    rectangle(100, 208, 460, 282);
    rectangle(100, 190, 280, 282);
    rectangle(100, 190, 460, 282);
    rectangle(100, 208, 190, 282);
    rectangle(100, 208, 370, 282);
    rectangle(100, 208, 460, 282);
    outtextxy(10, 230, ' Para' );
    outtextxy(10, 250, ' Klucz' );
    outtextxy(10, 270, ' Eskadra' );
    rectangle(9, 220, 460, 242);
    rectangle(9, 242, 460{550}, 262);
    rectangle(9, 262, 460, 282);
end;

Procedure wyp1;
begin
    for iio:=1 to 3 do
        begin
            il:=40;
            iy:=iy+20;
            for io:=1 to 4 do
                begin
                    il:=il+90;
                    outtextxy(il, iy, tabstr[iio, io]);
                end;
            end;
        end;
end;

```

```

end;

end;

setlinestyle(0,0,3);
end;
procedure oknodol;
begin
setcolor(15);
graphdefaults;
rectangle(9,249,602,302);
setviewport(10,250,600,300,clipon);
clearviewport;
end;

procedure okno_p;
begin
graphdefaults;
setviewport(440,240,getmaxx,getmaxy,clipon);
clearviewport;
rectangle(1,1,195,70);
end;

```

```

Procedure tekst;
begin
outtextxy(10,10,' Podaj czas ');
outtextxy(10,20,' odtwarzania got.boj. ');
outtextxy(10,30,' Dla: ');
rectangle(1,1,195,70);
end;

```

```

Procedure poprawa;
begin
clearviewport;
outtextxy(10,10,'Wpr. poprawna wartosc');
outtextxy(10,20,' czasow odtwarzania');
outtextxy(10,30,' got.boj. ');
outtextxy(10,50,' Nacisnij "E N T E R"');
rectangle(1,1,195,70);
begin
repeat
r:=readkey;

```

```

until r=#13;
graphdefaults;

end;
procedure min_b(a,b :integer; var ii:integer);
label
ss2,ss1,sss;

var

flaga,blad,i:integer;
p:char;
begin
setcolor(0);
bar(a,b-2,a+60,b+12);
line(a,b+10,a+50,b+10);
liczba:='';
i:=1;
flaga:=0;
repeat
begin
sss:    p:=readkey;
        if p=#8 then goto ss1;
        if p=#13 then goto ss2;
        if ( (p in ['0'..'9'])
or (p = ('-')) or (p=('+'))) then
            goto ss1 else
                begin
                    Sound(300);
                    Delay(100);
                    Nosound;
                    goto sss;
                end;
        begin
            liczba:=liczba+p;
            flaga:=1;
        end;
ss1:    if p=#8 then
        begin
            if ((flaga=1)and (i>2)). then
                if i>1 then

```

```

begin
  i:=i-1;
  if i=2 then flaga:=0;
  delete(liczba,i,1);
  Sound(600);
  Delay(100);
  Nosound;
end;
end
else
begin
if(((p in ) and ((p in ['0'..'9'])
or (p = ('-')) or (p=('+')))) then
begin
  liczba:=liczba+p;
  val(liczba,ii,blad);
  if ii>=999999 then
begin
  Sound(600);
  Delay(100);
  Nosound;
  i:=i-1;
  delete(liczba,i,1);
  goto sss;
end;

  i:=i+1;
end
else
begin
  if flaga=0 then begin
if (((i=1) or (i=2)) and (p='.')) then
begin
  liczba:=liczba+p;
  flaga:=1;
end;
end;
if (((p='.') and (flaga=1))and (i>2)) then
begin end

```

```

else
begin
i:=i+1;

end;
if ((i>1) and (p in['0'..'9'])) then
begin
liczba:=liczba+p;
val(liczba, ii, blad);
if ii>=999999 then
begin
Sound(600);
Delay(100);
Nosound;
i:=i-1;
delete(liczba, i, 1);
goto sss;
end;
end
else
begin
if p=#8 then
begin
if i>1 then
begin
i:=i-1;
delete(liczba, i, 1);
end
end
else
begin
end;
end;
end;

end;
end;
setcolor(0);
bar(a, b-2, a+60, b+12);
line(a, b+10, a+50, b+10);
OutTextXY(a, b, liczba);

```

```

    end;
    until (i>7) or ((p=#13)and(i>1));
ss2:   val(liczba,ii,blad);
setcolor(15);

end;

function int(a,b :integer) : integer;
var
    i,ii,blad: integer;
    p: char;
begin
    liczba:='';
    i:=1;
    repeat
        begin
            p:=readkey;
            if p=#8 then
                begin
                    if i>1 then
                        begin
                            i:=i-1;
                            delete(liczba,i,1);
                        end
                    else
                        begin
                            for i:= Sound(600);
                            Delay(100);
                            Nosound;
                            end;
                            then
                                end
                            begin
                                else
                                    SetFillStyle(1,1);
                                    begin
                                        bar(px,py+(1-n)*15,kx,ky+(1-n)*15);
                                        if p in ['0'..'9'] then
                                            begin
                                                Color(0);
                                                liczba:=liczba+p;
                                                i:=i+1;
                                            end
                                        else
                                            if (p<>#13)or((p=#13)and(i=1)) then
                                                begin

```

```

                Sound(400);
                Delay(100);
                Nosound;
            end;
        end;
        SetFillStyle(0,0);
        bar(a, b, a+30, b+10);
        OutTextXY(a, b, liczba);
        SetFillStyle(1,1);
    end;
until (i>4) or ((p=#13)and(i>1));
val(liczba, ii, blad);
int:=ii;
end;
procedure scroll1(il, px, py, kx, ky: integer;
                obiekt:tkom150; var k: integer);
var
i, m, n, z : integer;
begin
    SetFillStyle(0,0);
    k:=1;
    m:=1;
    if il<21 then n:=il
        else n:=21; {bylo 15}
        repeat
            for i:=m to n do
                begin
                    bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
                    if i=k then
                        begin
                            SetFillStyle(1,1);
                            bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
                            SetFillStyle(0,0);
                            SetColor(0);
                            outtextXY(px+4, py+4+(i-m)*15, obiekt[i]);
                            SetColor(1);
                        end
                    else
                        outtextXY(px+4, py+4+(i-m)*15, obiekt[i]);
                    end;
                end;
            end;
end;

```

```

OuttextXY(10,333,'WYBIERZ WARIANT I NACISNIJ  "Enter" ');
repeat
  r:=readkey;
  if r=#0 then
    begin
      r:=readkey;
      z:=0;
      case r of
        'G' : begin
          k:=1;
          m:=1;
          if il<15 then n:=il
            else n:=21;{15;}
          z:=1;
        end;
        'O' : begin
          k:=il;
          if il<21{15} then m:=1
            else m:=il-20; {14}
          n:=il;
          z:=1;
        end;
        'H' : begin
          k:=k-1;
          if k=0 then
            begin
              k:=1;
              Sound(800);
              Delay(300);
              Nosound;
            end
          else
            begin
              z:=1;
              if k<m then begin
                m:=m-1;
                n:=n-1;
              end
            else
              begin
                SetColor(1);
                outtextXY(px+4,py+4+(k-n)*15);
                SetColor(0);
              end
            end;
      end;
    end;
  end;

```

```

        z:=0;
        SetFillStyle(1,1);
        bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
        SetFillStyle(0,0);
        SetColor(0);
        outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
        SetColor(1);
        bar(px,py+(k-m+1)*15,kx,ky+(k-m+1)*15);

        outtextXY(px+4,py+4+(k-m+1)*15,obietk[k+1]);

        end;

    end;

    end;

    'P' : begin
        k:=k+1;
        if k>il then
            begin
                k:=il;
                Sound(800);
                Delay(300);
                Nosound;
            end
        else
            begin
                z:=1;
                if k>n then begin
                    n:=n+1;
                    m:=m+1;
                end
            else
                begin
                    z:=0;
                    SetFillStyle(1,1);
                    bar(px,py+(k-m)*15,kx,ky+(k-m)*15);

                    SetFillStyle(0,0);
                    SetColor(0);
                    outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
                    SetColor(1);

```

```

        bar(px, py+(k-m-1)*15, kx, ky+(k-m-1)*15);
    outtextXY(px+4, py+4+(k-m-1)*15, obiekt[k-1]);
        end;
    end;
end;
end
else
    if r<>#13 then
        begin
            Sound(800);
            Delay(300);
            Nosound;
        end;
    until (r=#13)or(z=1)
until r=#13;
    SetFillStyle(1,1);

end;
procedure scroll2(il, px, py, kx, ky: integer;
                obiekt:tkomwydr; var k: integer);
var
i, m, n, z : integer;
begin
    SetFillStyle(0,0);
    k:=1;
    m:=1;
    if il<21 then n:=il
        else n:=16; {bylo 15}
    repeat
    for i:=m to n do
    begin
        bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
        if i=k then
            begin
                SetFillStyle(1,1);
                bar(px, py+(i-m)*15, kx, ky+(i-m)*15);
                SetFillStyle(0,0);
                SetColor(0);
            end;
        k:=k+1;
        m:=m+1;
    end;
    until z=1;
end;

```

```

outtextXY(px+4,py+4+(i-m)*15, obiekt[i]);
    SetColor(1);
    end
else
outtextXY(px+4,py+4+(i-m)*15, obiekt[i]);
end;
repeat
r:=readkey;
if r=#0 then
begin
r:=readkey;
z:=0;
case r of
'G' : begin
k:=1;
m:=1;
if il<15 then n:=il
else n:=16; {15;}
z:=1;
end;
'O' : begin
k:=il;
if il<16{15} then m:=1
else m:=il-16; {14}
n:=il;
z:=1;
end;
'H' : begin
k:=k-1;
if k=0 then
begin
k:=1;
Sound(800);
Delay(300);
Nosound;
end
else
begin
z:=0;
SetFillStyle(z:=1);
bar(px,py+(k-1)*15);

```

```

m:=m-1;
n:=n-1;
end
else
begin
z:=0;
SetFillStyle(1,1);

bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
SetFillStyle(0,0);
SetColor(0);
outtextXY(px+4,py+4+(k-m)*15,obietk[k]);
SetColor(1);
bar(px,py+(k-m+1)*15,kx,ky+(k-m+1)*15);

outtextXY(px+4,py+4+(k-m+1)*15,obietk[k+1]);

end;
end;
end;
'P' : begin
k:=k+1;
if k>il then
begin
k:=il;
Sound(800);
Delay(300);
Nosound;
end
else
begin
z:=1;
if k>n then begin
n:=n+1;
m:=m+1;
end
else
begin
z:=0;
SetFillStyle(1,1);
bar(px,py+(k-m)*15,kx,ky+(k-m)*15);

```

```

        SetFillStyle(0,0);
        SetColor(0);
        outtextXY(px+4,py+4+(k-m)*15, obiekt[k]);
        SetColor(1);

        bar(px,py+(k-m-1)*15,kx,ky+(k-m-1)*15);
        outtextXY(px+4,py+4+(k-m-1)*15, obiekt[k-1]);
        end;
    end;
end;
else
    if r<>#13 then
        begin
            Sound(800);
            Delay(300);
            Nosound;
        end;
    until (r=#13)or(z=1)
until r=#13;
    SetFillStyle(1,1);
end;

procedure answer(a,c:integer; var f:integer);
var
w:string[1];
b,d,e:integer;
begin
    if a=1 then begin
        b:=20;
        d:=300;
        e:=260;
    end
    else

```

```

begin
    b:=60;
    d:=220;
    e:=250;
end;

str(c,w);
OutTextXY(b,d,' Podaj numer ( 1..' + w + ' ) : ');
    r:='n';
repeat
    f:=int(e,d);
    if (f>0)and(f<c+1) then r:='t'
    else
        begin
            Sound(800);
            Delay(300);
            Nosound;
        end;
    until r='t';
OuttextXY(20,320,'Do realizacji wybrano wariant');
OuttextXY(260,320,liczba);
    Moveto(265,333);
    Outtext(' ( Nacisnij ENTER)');
repeat
    r:=readkey;
until r=#13;
end;

begin

flag:=0;
ile:=0;
for s:=1 to 49 do
begin
    tdyz[s]:=0;
    tprzerwy[s]:=0;
    tabscr[s]:=' ';
end;

graphdefaults;
setviewport(1,50,getmaxx,getmaxy,clipon);

wybor:=iltabpocz-1;
tab_odtworz;

```

```

case wybor of
  1:begin

  outtextxy(10,200,'Ogolna liczba
  samolotow.w dyzurowaniu.....* ');
oknodol;

w1:outtextxy(20,20,'Ogolna liczba
samolotow w dyzurowaniu. ...');

  min_b(400,20,N);
  if ((N<=0) or (N>200)) then
    begin
      clearviewport;
      setcolor(0);
      bar(1,1,600,50);
outtextxy(20,20,'(Mozliwy zakres (1..200)
  Nacisnij ENTER');
      repeat
        r:=readkey;
        until r=#13;
      clearviewport;setcolor(15);goto w1;
      end

    else
      begin
        graphdefaults;
        setcolor(0);
        str(N:4,sstel);
        outtextxy(400,200,''+sstel);
        zmienne[17]:=sstel;
        end;

graphdefaults;
setcolor(15);
setviewport(8,248,603,303,clipon);
clearviewport;
graphdefaults;
setviewport(1,60,getmaxx,getmaxy,clipon);
tx:=tdyz1;
Tab_odtwarz;

```

```

iy:=210;
il:=40;
wyp1; {wypelnianie tabeli got.boj}
a:=iltypm;
ramki_tab_got;
outtextxy(470,180,'Czy zmieniasz dane ?');
rectangle(478,198,542,235);
scroll1(2,480,200,540,217,czyzm,taknie);

setlinestyle(0,0,1);
skok1:case taknie of
    1:begin
        okno_p;
        tekst;
        outtextxy(10,40,'PARY [min]');
        min_b(130,40,T1);
        a_1;
        okno_p;
        tekst;
        outtextxy(10,40,'KLUCZA [min]');
        min_b(130,40,T2);
        if T1>T2 then
            begin
                poprawa;
                goto skok1;
            end;
        graphdefaults;
        setcolor(0);
        bar(ppx,305,ppx+70,320);
        str(T2:3,ssss);
        outtextxy(ppx+20,310,+ssss);
        setcolor(15);
        okno_p;
        tekst;
        outtextxy(10,50,'ESKADRY [min]');
        min_b(130,40,T3);
        if T2>T3 then
            begin
                poprawa;
                goto skok1;
            end;
    end;

```

```

end;
graphdefaults;
setcolor(0);
bar(ppx, 325, ppx+70, 340);
str(T3:3, ssss);
outtextxy(ppx+20, 330, +ssss);
setcolor(15);
end;
2:begin {nie zmieniam danych - odczyt z tabel}
wybor_T1T2T3;
end;
end;
s:=1;
setviewport(450, 1, getmaxx, getmaxy, clipoff);
clearviewport;
rectangle(1, 1, 250, getmaxy-20);
for s:=1 to n do
begin
ng:=trunc(n/s);
if s<=2 then
begin
T0:=t1;
Tx1:=tx;
end
else
begin
T0:=((T3-T2)*(S-4)/14)+T2;
Tx1:=0.97*Tx;
end;
if T0<((ng-1)*TX1-2*TD-4-(S-1)/3) then
begin
tprzerwy[s]:=0;
setviewport(450, 1, getmaxx, getmaxy, clipoff);
end
else
begin

```

```

t:=Ng*Tx1;
tp:=abs(t0-(NG-1)*Tx1+2*TD+4+(s-1)/3);
tdyz[s]:=t;
tprzerwy[s]:=tp;
end;

end;
outtextxy(25,10,'S ');
outtextxy(25,20,'K Czas Czas ');
outtextxy(25,30,'L ciagl. przerwy ');
outtextxy(25,40,'A dyzur. ');
outtextxy(25,50,'D [ min ] [ min ]');
iy:=60;

for s:=1 to n do
begin
iy:=iy+10;
str(s:3,ss);
str(tdyz[s]:5:2,st);
str(tprzerwy[s]:5:2,stp);
if n>25 then begin
if tprzerwy[s]=0 then
begin
tabscr[s]:=ss+' Dyz. ciagle';
end
else
begin
tabscr[s]:=ss+' '+st+' '+stp;
end;
end
else
begin
if tprzerwy[s]=0 then
begin
outtextxy(10,iy,ss+' Dyz. ciagle');
end
else
outtextxy(10,iy,ss+' '+st+' '+stp);
end;
end;
end;

```

```

end;
RECTANGLE(10,65,230,iy+15);
rectangle(10,5,40,iy+15);
rectangle(10,5,230,60);
if n>25 then
scroll12(49,15,80,220,96,tabscr,ilek);
repeat until keypressed;
clearviewport;
outtextxy(10,20,'Nacisnij ENTER');

repeat
r:=readkey;
until r=#13;
outtextxy(10,100,'Czy powtarzasz');
outtextxy(10,110,'obliczenia ?');
rectangle(8,128,162,164);
scroll11(2,10,130,160,147,czyzm,kut);
if kut=1 then flag:=1
else flag:=2;
goto kkk;

setviewport(1,1,getmaxx,getmaxy,clipon);
goto pocz;
end;

2:begin {początek iltabpocz=3 czyli wybor=2}

outtextxy(10,200,'Skład grupy
dyzurującej w strefie ( N < 12 ) * ');
setcolor(0);
bar(10,210,620,240);
setcolor(15);
oknodol;

w21:outtextxy(20,20,'Skład grupy
dyzurującej w strefie (N<12) ');

```

```

min_b(400,20,S);
if ((S<=0) or (S>12)) then
    begin
        clearviewport;
        setcolor(0);
        bar(1,1,600,50);
    outtextxy(20,20,'(Mozliwy zakres (1..12) Nacisnij ENTER)');
        repeat
            r:=readkey;
            until r=#13;
        clearviewport;setcolor(15);goto w21;
        end
    else
        begin
            graphdefaults;
            setcolor(0);
            str(S:4,sstel);
            outtextxy(400,200,''+sstel);
            zmienne[19]:=sstel;
            end;
        graphdefaults;
        setcolor(15);
        setviewport(8,248,603,303,clipon);
        clearviewport;
        oknodol;
    outtextxy(10,10,'Przez jaki okres czasu
    masz dyzurowac');
        rectangle(318,18,582,55);
        scroll1(2,320,20,580,37,jakidyz,ob);
        case ob of
            1:begin
                graphdefaults;
                setcolor(0);
                outtextxy(10,220,'Dyzurowanie w czasie
                nieograniczonym ');
                setcolor(15);
                end;
            2:begin
                oknodol;

```

```

clearviewport;
outtextxy(10,20,'Czas dyzowania [min]');
min_b(200,20,TT);
str(TT:5,sstel);
graphdefaults;
setcolor(0);
outtextxy(10,220,'Dyzuowanie w czasie [min]....
.....* '+sstel);
setcolor(15);
end;
end;
end;
end;
graphdefaults;
setcolor(15);
setviewport(8,248,603,303,clipon);
clearviewport;
graphdefaults;
setviewport(1,60,getmaxx,getmaxy,clipon);

tx:=tdyz1;

iy:=210;
il:=40;

a:=iltypm;
{ramki_tab_got;}

wybor_T1T2T3;
if s<=2 then
begin
T0:=t1;
Tx1:=tx;
end
else
begin
T0:=((T3-T2)*(S-4)/14)+T2;
TX1:=0.97*TX;
end
end;

```

```

oknodol;
setcolor(15);
str(S:3,sN);
str(T0:4:0,sstel);
outtextxy(10,10,'Sredni czas
                odtwarzania gotowosci'+sN+' samolotow');
outtextxy(10,20,'    wynosi          '+sstel+' min');

graphdefaults;

okno_p;
                outtextxy(10,10,'Czy zmieniasz czas ');
outtextxy(10,20,'odtworzania got.boj. ? ');
                rectangle(1,1,195,70);
                rectangle(38,38,152,75);
                scroll1(2,40,40,150,57,czyzm,taknie);
setlinestyle(0,0,1);

skok31:case taknie of
    1: begin
        okno_p;
        tekst;
        outtextxy(10,40,'GRUPY [min]');
        min_b(130,40,T01);
        setcolor(15);
    end;
    2: begin

        end;
    end;
r:=readkey;
if taknie=1 then T0:=T01;
pocz:outtextxy(10,80,'Czy nalezy:');
{Poczatek obliczen}
                outtextxy(10,100,'obliczenia');
ng:=((T0+2*Td+4+(S-1)/3)/Tx1)+1;
if ob=1 then scroll(130,200,147,czyzm,kut);
if kut=1 then begin
else flag:=2;
clearviewport; end

```

```

else
begin
  if ng>(tt*s)/tx1 then
    begin
      ng:=(tt*s)/tx1;
    end;
  end;

if (ng-trunc(ng))=0 then
  begin
  end
else
  begin
    ng:=trunc(ng+1);
  end;

n1:=trunc(ng*s);

setviewport(450,1,635,320,clipoff);

clearviewport;
rectangle(1,1,190,340);
outtextxy(10,10,'Dla zapewnienia nakazanych');
outtextxy(10,20,'warunkow dyzurowania ');
outtextxy(10,30,'potrzeba nie mniej niz ');
str(n1:4:0,sn);
outtextxy(10,40,sn+' samolotow');

druklin:
repeat
r:=readkey;
until r=#13;
pocz:outtextxy(10,80,'Czy nalezy:');
  outtextxy(10,90,' powtorzyc');
  outtextxy(10,100,' obliczenia');
rectangle(8,128,202,164);
scroll1(2,10,130,200,147,czym,kut);
if kut=1 then flag:=1
else flag:=2;
clearviewport;

```



```

    unit danesys;
interface
uses
crt, dos, graph;
var
smin1:string[1];
ssm, sssr, sssd:string[5];
j, driver, mode: integer;
r: char;
i, drop: word;
day, year, month, dayofweek: word;
hour, min, sec, sec100: word;
shour, smin, ssec, ssec100: string[6];
const
days:array[0..6]of string[10]=
('Niedziela ',
 'Poniedz.  ',
 'Wtorek    ',
 'Sroda     ',
 'Czwartek ',
 'Piatek    ',
 'Sobota    ');

```

```

Procedure dane_zmienne;
implementation

```

```

Procedure dane_zmienne;

```

```

procedure litera_P(x,y: integer);
begin
line(x,y+15,x,y);
line(x+2,y+15,x+2,y);
line(x+2,y+15,x+2,y+30);
moveto(x,y{50,15});
lineto(x+25,y);
lineto(x+25,y+1);
lineto(x+25,y+15);
lineto(x,y+15); lineto(x,y+30);

```

```

end;

procedure litera_R(x,y: integer);
begin
line(x,y+15,x,y);
line(x+2,y+15,x+2,y);
line(x+2,y+15,x+2,y+30);
moveto(x,y);
lineto(x+25,y);
lineto(x+25,y+15); lineto(x,y+15); lineto(x,y+30);
moveto(x,y+15);
lineto(x+23,y+30);
end;

```

```

procedure litera_I(x,y: integer);
begin
line(x,y+15,x,y);
rectangle(x,y,x+5,y+30);
end;

```

```

procedure myslnik(x,y: integer);
begin
rectangle(x,y+15,x+30,y+20);
end;

begin
{driver:=5;}
nosound;
rectangle(1,1,getmaxx-1,getmaxy-1);
rectangle(2,2,getmaxx-2,getmaxy-2);
setcolor(0);
bar(10,10,250,60);
rectangle(12,12,248,58);
outtextxy(20,20,'AKADEMIA OBRONY NARODOWEJ');
Outtextxy(20,30,'Wydział Wojsk Lotniczych ');
outtextxy(20,40,' i Obrony Powietrznej ');
{litera_p(50,15);

```

```

litera_r(85,15);
litera_I(120,15);
litera_R(140,15);}
{myslnik(190,15);}
setcolor(15);
getdate(drop,drop,drop,day);
getdate(year,month,day,dayofweek);
str(year:5,sssr);
str(day:2,sssd);
if sssd<='9' then sssd:='0'+sssd;
str(month:2,ssm);

outtextxy(300,20,sssd+'/' +ssm+'/' +sssr+'
- '+days[dayofweek]);

rectangle(5,90,getmaxx-5,130);
setcolor(0);
bar(10,100,getmaxx-10,120);
outtextxy(20,110,'...Program w trakcie testowanie
Program DYZUR,. DYZUR ,,,,,,,,,,,,,,,,,,,,,,');
setcolor(15);
gettime(hour,min,sec,sec100);
str(hour:5,shour);
str(min:2,smin);
str(sec:2,ssec);
outtextxy(330,30,shour+' ':' +smin);
setcolor(15);
rectangle(1,290,getmaxx-1,getmaxy);
outtextxy(10,300,'Koncepcja, algorytm,
program : mjr nawig. dr Piotr MAKOWSKI');
outtextxy(10,310,mjr dr inz. Andrzej GRZELKA');

i:=100;
for j:=1 to 20
do begin
    delay(50);
    i:=i+50;
    sound(i);

end;
```

```

nosound;
repeat
r:=readkey;
until r=' ';
type A, B, C, G, H, I, J;
end;
end.
array(1..30) of string(30);
k:=array(1..30) of string(30);
var
H1:real;
L1, L2, L3, L4, L5:real;
L10, L11, L12, L13, L14, L15:real;
L16, L17, L18, L19, L20:real;
L21, L22, L23, L24, L25:real;
L26, L27, L28, L29, L30:real;
L31, L32, L33, L34, L35:real;
L36, L37, L38, L39, L40:real;
L41, L42, L43, L44, L45:real;
L46, L47, L48, L49, L50:real;
L51, L52, L53, L54, L55:real;
L56, L57, L58, L59, L60:real;
L61, L62, L63, L64, L65:real;
L66, L67, L68, L69, L70:real;
L71, L72, L73, L74, L75:real;
L76, L77, L78, L79, L80:real;
L81, L82, L83, L84, L85:real;
L86, L87, L88, L89, L90:real;
L91, L92, L93, L94, L95:real;
L96, L97, L98, L99, L100:real;
Woc, N1, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10: integer;
dy, ast: real;
zlaw, zosc: integer;
Procedure maksymalizujlog21(Woc: integer; N1,
TEM: real; var Vakk: real);
Procedure OBLICZ_CK21b(N1, Vp, ast: real; Woc:
integer; var ck: real);
Procedure zlaw21b(N1, Vp, Vakk, Y3: real;
var ck, Tz: real);
Procedure zosc21b(N1, Vp: real; Woc: integer;
var ast, Qp, Sz2a, Qz2a, Tz2a: real);
Procedure zlaw21b(N1, Vp: real; var Tz2a, Qz2a, Sz2a: real);
Implementation
Procedure maksymalizujlog21(Woc: integer; N1, TEM: real;
var Vakk: real);
begin
case Woc of
9..11: Vp:=700-0.018*N1;
12..13: Vp:=680-0.016*N1;
14..15: Vp:=670-0.015*N1;
16..20: Vp:=680-0.014*N1; {przrzadowa}
end;
Vakk:=ln(Vp)+0.5*ln(TEM)-2.831-2.628*ln(1-0.0000226*N1);
Vnkk:=exp(Vakk); {rzeczywista w km/h}

```

```

unit dyz_21;
interface
uses
  typy_n, dos, graph, crt;
type
tkom29=array[1..30]of string[30];
tkom=array[1..32] of string[53];
var
  iii:real;
  tel, ii: longint;
  liczba:string[5];
  r:char;
      Warpopraw:tkom29;
  woc, moo, p1, p2, tk, zb{, kpopraw}: integer;
  dy, mst:real;
  zlawartosc: integer;
  Procedure maksymalnyzasieg21(Woc: integer; H1,
  TEM:real; var Vmkk:real);
  Procedure OBLICZ_CK21b(H1, Vp, mst:real; woc:
  integer; var ck:real);
  Procedure manewr21b(H1, Vp, Vmkk, V3:real;
  var ck, Tm:real);
  Procedure wznoszenie21b(Hkk, Hp:real; Woc: integer;
  var mst, Qp, Swzn, Qwzn, Twzn:real);
  Procedure znizanie21b(Hkk, Hp:real; var Tzn, Qzn, Szn:real);
implementation

Procedure maksymalnyzasieg21(Woc: integer; H1, TEM: real;
      var Vmkk:real);

begin
  case Woc of
    9..11: Vp:=700-0.018*H1;
    12..13: Vp:=680-0.016*H1;
    14..15: Vp:=670-0.015*H1;
    16..20: Vp:=660-0.014*H1; {przyczadowa}
  end;

  Vmkk:=ln(Vp)+0.5*ln(TEM)-2.831-2.628*ln(1-0.000226*H1);
  Vmkk:=exp(Vmkk); {rzeczywista w km/h}

```

```

end;

Procedure OBLICZ_CK21b(H1, Vp, mst: real; woc: integer;
                      var ck: real);
var
    flaga, W: integer;
    ck1, ck2, hh1, hh2, m, x, aa, hh, A, B, C: real;
    label
    skok;

begin
    case trunc(H1) of
        200..500: W:=1;
        501..11000: W:=trunc(H1*0.001)+2;
    end;
    flaga:=1;
    x:=Vp/100;
    skok: case W of
        1: begin
            A:=-0.3657*(x-8)*(x-9)*(x-10.8)+
                1.0142*(x-7)*(x-9)*(x-10.8)+
                -0.8472*(x-7)*(x-8)*(x-10.8)+
                0.2297*(x-7)*(x-8)*(x-9.0);

            B:=-0.3010*(x-8)*(x-9)*(x-10.6)+
                1.0256*(x-6.8)*(x-9)*(x-10.6)+
                -0.9659*(x-6.8)*(x-8)*(x-10.6)+
                0.2846*(x-6.8)*(x-8)*(x-9.0);

            C:=-0.2736*(x-8)*(x-9)*(x-10.2)+
                1.1201*(x-6.6)*(x-9)*(x-10.2)+
                -1.3194*(x-6.6)*(x-8)*(x-10.2)+
                0.4892*(x-6.6)*(x-8)*(x-9.0);

            AA:=0.0071;
            HH:=200;
        end;
    end;
end;

```

```

end;
2:begin
  A:=-0.2985*(x-8)*(x-9)*(x-10.8)+
    0.8961*(x-6.9)*(x-9)*(x-10.8)+
    -0.7936*(x-6.9)*(x-8)*(x-10.8)+
    0.2187*(x-6.9)*(x-8)*(x-9.0);

  B:=-0.3072*(x-8)*(x-9)*(x-10.4)+
    1.0555*(x-6.8)*(x-9)*(x-10.4)+
    -1.0876*(x-6.8)*(x-8)*(x-10.4)+
    0.3637*(x-6.8)*(x-8)*(x-9.0);

  C:=-0.2809*(x-8)*(x-9)*(x-10.0)+
    1.2071*(x-6.6)*(x-9)*(x-10.0)+
    -1.5624*(x-6.6)*(x-8)*(x-10.0)+
    0.6617*(x-6.6)*(x-8)*(x-9.0);

  AA:=0.0079;
  HH:=500;

```

```

end;
3:begin
  A:=-0.2714*(x-8)*(x-9)*(x-10.4)+
    0.9270*(x-6.8)*(x-9)*(x-10.4)+
    -0.9740*(x-6.8)*(x-8)*(x-10.4)+
    0.3224*(x-6.8)*(x-8)*(x-9.0);

  B:=-0.2314*(x-8)*(x-9)*(x-10.2)+
    0.9577*(x-6.6)*(x-9)*(x-10.2)+
    -1.1597*(x-6.6)*(x-8)*(x-10.2)+
    0.4419*(x-6.6)*(x-8)*(x-9.0);

  C:=-0.2170*(x-8)*(x-9)*(x-9.8)+
    1.1388*(x-6.4)*(x-9)*(x-9.8)+
    -1.7884*(x-6.4)*(x-8)*(x-9.8)+
    0.8476*(x-6.4)*(x-8)*(x-9.0);

  AA:=0.0133;
  HH:=1000;

```

```

end;
4:begin
  A:=-0.2746*(x-8)*(x-9)*(x-10.0)+
    1.0166*(x-6.8)*(x-9)*(x-10.0)+
    -1.2636*(x-6.8)*(x-8)*(x-10.0)+
    0.5936*(x-6.8)*(x-8)*(x-9.0);

  B:=-0.2353*(x-8)*(x-9)*(x-9.8)+
    1.0714*(x-6.6)*(x-9)*(x-9.8)+
    -1.6406*(x-6.6)*(x-8)*(x-9.8)+
    0.8138*(x-6.6)*(x-8)*(x-9.0);

  {?} C:=-0.2446*(x-8)*(x-8.9)*(x-9.2)+
    1.7361*(x-6.4)*(x-8.9)*(x-9.2)+
    -5.0370*(x-6.4)*(x-8)*(x-9.2)+
    3.7698*(x-6.4)*(x-8)*(x-8.9);

    AA:=0.0277;
    HH:=2000;

end;
5:begin
  A:=-0.2083*(x-8)*(x-9)*(x-9.6)+
    1.0267*(x-6.6)*(x-9)*(x-9.6)+
    -1.9444*(x-6.6)*(x-8)*(x-9.6)+
    1.2152*(x-6.6)*(x-8)*(x-9.0);

  B:=-0.4174*(x-7.5)*(x-8.4)*(x-9.4)+
    1.4035*(x-6.5)*(x-8.4)*(x-9.4)+
    -1.6023*(x-6.5)*(x-7.5)*(x-9.4)+
    0.6715*(x-6.5)*(x-7.5)*(x-8.4);

  C:=-0.4405*(x-7.5)*(x-8.4)*(x-9.0)+
    1.7912*(x-6.4)*(x-8.4)*(x-9.0)+
    -2.8425*(x-6.4)*(x-7.5)*(x-9.0)+
    1.5512*(x-6.4)*(x-7.5)*(x-8.4);

    AA:=0.0428;
    HH:=3000;

```

```

end;
6: begin
  A:=-0.9523*(x-7.2)*(x-8)*(x-9)+
    2.2916*(x-6.6)*(x-8)*(x-9)+
    -1.9642*(x-6.6)*(x-7.2)*(x-9.0)+
    0.6712*(x-6.6)*(x-7.2)*(x-8.0);

  B:=-0.6868*(x-7.2)*(x-8.0)*(x-8.8)+
    2.1484*(x-6.4)*(x-8.0)*(x-8.8)+
    -2.3925*(x-6.4)*(x-7.2)*(x-8.8)+
    1.0091*(x-6.4)*(x-7.2)*(x-8.0);

  C:=-0.5393*(x-7.2)*(x-8.0)*(x-8.6)+
    2.1517*(x-6.2)*(x-8.0)*(x-8.6)+
    -3.2407*(x-6.2)*(x-7.2)*(x-8.6)+
    1.6369*(x-6.2)*(x-7.2)*(x-8.0);

    AA:=0.0571;
    HH:=4000;

end;
7: begin
  A:=-0.5150*(x-7)*(x-8)*(x-8.6)+
    1.4453*(x-6.2)*(x-8)*(x-8.6)+
    -1.9907*(x-6.2)*(x-7)*(x-8.6)+
    1.1718*(x-6.2)*(x-7.0)*(x-8.0);

  B:=-0.5078*(x-6.8)*(x-8.0)*(x-8.4)+
    1.3020*(x-6.0)*(x-8.0)*(x-8.4)+
    -2.5625*(x-6.0)*(x-6.8)*(x-8.4)+
    1.7903*(x-6.0)*(x-6.8)*(x-8.0);

  C:=-0.5429*(x-6.8)*(x-7.6)*(x-8.0)+
    2.3437*(x-5.8)*(x-7.6)*(x-8.0)+
    -4.5138*(x-5.8)*(x-6.8)*(x-8.0)+
    2.8409*(x-5.8)*(x-6.8)*(x-7.6);

end;
10: begin
  AA:=0.0699;
  HH:=5000;

```

```

end;
8:begin
  A:= 0.9545*(x-6.8)*(x-8.2)+
    -1.5446*(x-6.0)*(x-8.2)+
    0.8441*(x-6.0)*(x-6.8);

  B:= 0.9629*(x-6.8)*(x-8.0)+
    -1.7592*(x-5.9)*(x-8.0)+
    1.0515*(x-5.9)*(x-6.8);

  C:= 1.0000*(x-6.8)*(x-7.8)+
    -2.1299*(x-5.8)*(x-7.8)+
    1.3499*(x-5.8)*(x-6.8);

  AA:=0.0875;
  HH:=6000;

end;
9:begin
  A:= 1.0555*(x-6.6)*(x-7.6)+
    -2.0000*(x-5.8)*(x-7.6)+
    1.2611*(x-5.8)*(x-6.6);

  B:= 1.1111*(x-6.6)*(x-7.4)+
    -2.5138*(x-5.7)*(x-7.4)+
    1.6911*(x-5.7)*(x-6.6);

  C:= 1.1749*(x-6.6)*(x-7.2)+
    -3.4166*(x-5.6)*(x-7.2)+
    2.5520*(x-5.6)*(x-6.6);

end;
12:begin
  AA:=0.1000;
  HH:=7000;

end;
10:begin
  A:= 1.7261*(x-6.2)*(x-7.0)+

```

$$\begin{aligned} & -3.1458*(x-5.6)*(x-7.0)+ \\ & 1.6964*(x-5.6)*(x-6.2); \end{aligned}$$

$$\begin{aligned} B:= & 1.7582*(x-6.2)*(x-6.8)+ \\ & -3.9285*(x-5.5)*(x-6.8)+ \\ & 2.5641*(x-5.5)*(x-6.2); \end{aligned}$$

$$\begin{aligned} C:= & 1.5982*(x-6.2)*(x-6.8)+ \\ & -3.9583*(x-5.4)*(x-6.8)+ \\ & 2.5000*(x-5.4)*(x-6.2); \end{aligned}$$

$$AA:=0.1124;$$

$$HH:=8000;$$

end;

11:begin

$$\begin{aligned} A:= & 2.0138*(x-6.0)*(x-6.6)+ \\ & -4.2499*(x-5.4)*(x-6.6)+ \\ & 2.4999*(x-5.4)*(x-6.0); \end{aligned}$$

$$\begin{aligned} B:= & 2.6666*(x-6.0)*(x-6.4)+ \\ & -7.0833*(x-5.4)*(x-6.4)+ \\ & 4.7500*(x-5.4)*(x-6.0); \end{aligned}$$

$$\begin{aligned} C:= & 2.9166*(x-6.0)*(x-6.4)+ \\ & -7.4166*(x-5.4)*(x-6.4)+ \\ & 5.0000*(x-5.4)*(x-6.0); \end{aligned}$$

$$AA:=0.1250;$$

$$HH:=9000;$$

end;

12:begin

$$\begin{aligned} A:= & 3.2499*(x-5.6)*(x-6.2)+ \\ & -5.5416*(x-5.2)*(x-6.2)+ \\ & 2.7500*(x-5.2)*(x-5.6); \end{aligned}$$

$$B:= 4.5312*(x-5.6)*(x-6.0)+$$

```

-9.3749*(x-5.2)*(x-6.0)+
5.3124*(x-5.2)*(x-5.6);
{ C: ? }
AA:=0.1500;
HH:=10000;

end;

end;

m:=Mst*0.001;
AA:=AA*(m-7)+1;
if Woc<13 then
ck:=AA*(0.25*(B-A)*(Woc-9)+A)
else
ck:=AA*(0.25*(C-B)*(Woc-13)+B);
If flaga=1 then
begin
ck1:=ck; HH1:=HH; W:=W+1; flaga:=2;
goto skok;
end
else
begin
ck2:=ck; HH2:=HH;
end;
ck:=((ck2-ck1)/(HH2-HH1))*(H1-HH1)+ck1;
end;

```

```

Procedure manewr21b(H1, Vp, Vmkk, V3: real;

```

```

var ck, Tm: real);

```

```

var

```

```

flaga, W: integer;

```

```

y1, y2, y, x, A, B: real;

```

```

label

```

```

skok_H1, skok_H11;

```

```

BEGIN

```

```

if H1<=5000 then begin

```

```

if H1<3000 then x:=0.01*Vp; flaga:=1;

```

```

case trunc(H1) of

```

```

0..2999: W:=1;

```

```

3000..5000: W:=2;

```

```

        end;
    skok_H1:
    case trunc(W) of
        1:begin
            A:=8.125*(x-7)*(x-9)+
              -35.75*(x-5)*(x-9)+
              31.25*(x-5)*(x-7);
            B:=11.875*(x-7)*(x-9)+
              -42.50*(x-5)*(x-9)+
              37.50*(x-5)*(x-7);
            end;
        2:begin
            A:=21.428*(x-6.4)*(x-8)+
              -53.57*(x-5)*(x-8)+
              40.625*(x-5)*(x-6.4);
            B:=34.524*(x-6.4)*(x-8)+
              -93.75*(x-5)*(x-8)+
              67.71*(x-5)*(x-6.4);
            end;
        3:begin
            A:=42.5*(x-6)*(x-7)+
              -110.0*(x-5)*(x-7)+
              75.00*(x-5)*(x-6);
            B:=90.0*(x-6)*(x-7)+
              -250.00*(x-5)*(x-7)+
              190.00*(x-5)*(x-6);
            end;
    end;
    end;
    y:=((B-A)/7)*(Woc-9)+A;
    if flaga=1 then
        begin
            flaga:=2; W:=W+1; y1:=y; goto skok_H1;
        end
    else y2:=y;
    if H1<3000 then
        y:=((y2-y1)/2500)*(H1-500)+y1
    else
        y:=((y2-y1)/2000)*(H1-3000)+y1;

```

```

ck:=3.25*y/(V3*V3);
end
else
begin
flaga:=1; x:=Vmkk;
W:=trunc(0.0005*(H1-500))+1;
skok_H11:
case W of
1: y:=3.888*(x-0.7)*(x-1)+ {5000}
-9.111*(x-0.4)*(x-1)+
5.555*(x-0.4)*(x-0.7);
2: y:=3.166*(x-0.7)*(x-1)+ {7000}
-7.333*(x-0.4)*(x-1)+
4.499*(x-0.4)*(x-0.7);
3: y:=2.499*(x-0.7)*(x-1)+ {9000}
-5.555*(x-0.4)*(x-1)+
3.611*(x-0.4)*(x-0.7);
4: y:=1.944*(x-0.7)*(x-1)+ {11000}
-4.333*(x-0.4)*(x-1)+
2.777*(x-0.4)*(x-0.7);
end;
if flaga=1 then
begin
flaga:=2; W:=W+1; y1:=y;
goto skok_H11;
end
else y2:=y;
y:=0.0005*(y2-y1)*(H1-2000*(W-2)-500)+y1;
ck:=3600*y/V3;
end; b(Hkk,up:real;
Wcc:integer; var Tm:=0.308*V3; Szn,Tzn:real);
var END;

flaga,w:integer;
Procedure znizanie21b(Hkk,Hp:real; S,x,T,H:real;
label var Tzn,Qzn,Szn:real);

```

var

{ponizej lokalne}

{\*\*\*\*\*}

x:real;

BEGIN

{czas sek}

x:=0.001\*Hkk;

Tzn:=-0.1999\*(x-1)\*(x-11)+0.1799\*(x-1)\*(x-6);

x:=0.001\*Hp;

Tzn:=(-0.1999\*(x-1)\*(x-11)+0.1799\*(x-1)\*(x-6))-Tzn;

Tzn:=60\*Tzn;

{paliwo kg}

x:=0.001\*Hkk;

Qzn:=-1.9199\*(x-1)\*(x-11)+1.5599\*(x-1)\*(x-6);

x:=0.001\*Hp;

Qzn:=(-1.9199\*(x-1)\*(x-11)+1.5599\*(x-1)\*(x-6))-Qzn;

{droga km}

x:=0.001\*Hkk;

Szn:=10\*(x-1);

x:=0.001\*Hp;

Szn:=10\*(x-1)-Szn;

END;

Procedure wznoszenie21b(Hkk, Hp: real;

Woc: integer; var mst, Qp, Swzn, Qwzn, Twzn: real);

var

flaga, w: integer;

Q1, Q2, Q, HH, HH1, HH2, S1, S2, T1, T2, A, B, S, x, T, H: real;

label

```
skok_t, skok_s,  
skok_Q;
```

```
        BEGIN  
        {czas}  
        H:=Hkk;  
        case trunc(H) of  
            0..199: W:=0;  
            200..500: W:=1;  
            501..11000: W:=trunc(H*0.001)+2;  
        end;  
        flaga:=1;  
        x:=WOC;  
        skok_t:  
            case W of  
                0: begin  
                    T:=0; A:=0; B:=0; HH:=0;  
                end;  
                1: begin  
                    T:=0.0312*(x-9)+0.75;    A:=0;    B:=0;    HH:=200;  
                end;  
                2: begin  
                    T:=0.0312*(x-9)+1.05;    A:=0;    B:=0;    HH:=500;  
                end;  
                3: begin  
                    T:=0.0437*(x-9)+1.30;    A:=0;    B:=0.05;    HH:=1000;  
                end;  
                4: begin  
                    T:=0.0750*(x-9)+1.65;    A:=0;    B:=0.1;    HH:=2000;  
                end;  
                5: begin  
                    T:=0.1124*(x-9)+2.00;    A:=0.2;    B:=0.25;    HH:=3000;  
                end;  
                6: begin  
                    T:=0.1437*(x-9)+2.00;    A:=0.2;    B:=0.5;    HH:=4000;  
                end;  
                7: begin  
                    T:=0.0625*(x-15)*(x-17)+  
                    -0.3333*(x-9)*(x-17)+
```

```

0.2906*(x-9)*(x-15);

A:=0.2; B:=0.6; HH:=5000;
end;

8:begin
T:=0.0729*(x-15)*(x-17)+
-0.4166*(x-9)*(x-17)+
0.3593*(x-9)*(x-15);

A:=0.3; B:=0.9; HH:=6000;
end;

9:begin
T:=0.0874*(x-15)*(x-17)+
-0.5000*(x-9)*(x-17)+
0.4531*(x-9)*(x-15);

A:=0.3; B:=1.25; HH:=7000;
end;

10:begin
T:=0.1020*(x-15)*(x-17)+
-0.6250*(x-9)*(x-17)+
0.5630*(x-9)*(x-15);

A:=0.5; B:=3; HH:=8000;
end;

11:begin
T:=0.1210*(x-15)*(x-17)+
-0.7920*(x-9)*(x-17)+
0.7190*(x-9)*(x-15);

A:=0.7; B:=4.5; HH:=9000;
end;

12:begin
T:=0.8333*(x-9)+7; A:=1; B:=6.5; HH:=10000;

```

```

end;
13:begin
T:=0.917*(x-9)+8.75; A:=1.2; B:=9; HH:=11000;

end;

end;
T:=60*T;
T:=T+6*TE*(((B-A)/8)*(x-9)+A); {w sek}
if flaga=1 then begin
T1:=T; W:=W+1; HH1:=HH; flaga:=2; goto skok_t;
end
else
begin
T2:=T; HH2:=HH;
end;

if H=Hkk then
begin
Twzn:=((T2-T1)/(HH2-HH1))*(H-HH1)+T1;
flaga:=1;
W:=W-1; H:=Hp;
goto skok_t;
end
else
begin
Twzn:=Twzn-((T2-T1)/(HH2-HH1))*(H-HH1)-T1;
end;

{paliwo}
H:=Hkk;
case trunc(H) of
0..199: W:=0;
200..500: W:=1;
501..11000: W:=trunc(H*0.001)+2;
end;
flaga:=1;
x:=WOC;
skok_Q:

```

```

case W of
  0:begin
    Q:=0; A:=0; B:=0;HH:=0;
  end;
  1:begin
    Q:=2.25*(x-9)+140; A:=0; B:=0; HH:=200;
  end;
  2:begin
    Q:=3.00*(x-9)+170; A:=0; B:=0; HH:=500;
  end;
  3:begin
    Q:=4.50*(x-9)+188; A:=0; B:=0; HH:=1000;
  end;
  4:begin
    Q:=7.00*(x-9)+210; A:=3; B:=5; HH:=2000;
  end;
  5:begin
    Q:=10*(x-9)+230; A:=8; B:=10; HH:=3000;
  end;
  6:begin
    Q:=11.50*(x-9)+258; A:=10; B:=12; HH:=4000;
  end;
  7:begin
    Q:=14.25*(x-9)+280; A:=10; B:=15; HH:=5000;
  end;
  8:begin
    Q:=6.4170*(x-15)*(x-17)+
      -33.330*(x-9)*(x-17)+
      27.813*(x-9)*(x-15);
    A:=10; B:=16; HH:=6000;
  end;
  9:begin
    Q:=6.875*(x-15)*(x-17)+
      -36.6666*(x-9)*(x-17)+
      31.2500*(x-9)*(x-15);
    A:=15; B:=30; HH:=7000;
  end;
end;

```

```

end;

10:begin
  Q:=7.438*(x-15)*(x-17)+
    -41.6666*(x-9)*(x-17)+
    35.625*(x-9)*(x-15);

  A:=18; B:=30;          HH:=8000;
end;

11:begin
  Q:=8.125*(x-15)*(x-17)+
    -46.6666*(x-9)*(x-17)+
    40.3125*(x-9)*(x-15);

  A:=19; B:=45;          HH:=9000;
end;

12:begin
Q:=33.333*(x-9)+420; A:=28; B:=55; HH:=10000;

end;

13:begin
Q:=43.750*(x-9)+465; A:=40; B:=65; HH:=11000;

end;

end;

Q:=Q+0.1*TE*(((B-A)/8)*(x-9)+A); {w kg}
if flaga=1 then begin
Q1:=Q; W:=W+1; HH1:=HH; flaga:=2; goto skok_Q;
end
else
begin
Q2:=Q; HH2:=HH;
end;

if H=Hkk then
begin

```

```

Qwzn:=((Q2-Q1)/(HH2-HH1))*(H-HH1)+Q1;
        flaga:=1;
        W:=W-1; H:=Hp;
        goto skok_Q;
    end
else
    begin
Qwzn:=Qwzn-((Q2-Q1)/(HH2-HH1))*(H-HH1)-Q1;
        end;

{droga}
H:=Hkk;
case trunc(H) of
    0..1499:W:=0;

1500..11000:W:=trunc((H-500)*0.001);
end;
flaga:=1;
x:=WOC;
skok_S:
    case W of
        0:begin
            S:=0; A:=0; B:=0;HH:=0;
        end;
        1:begin
S:=0.3*(x-9)+2.0;   A:=0;   B:=0;   HH:=1500;
        end;
        2:begin
S:=0.688*(x-9)+7.5;   A:=1;   B:=2;   HH:=2500;
        end;

        3:begin
            S:=0.2910*(x-15)*(x-17)+
            -1.6660*(x-9)*(x-17)+
            1.500*(x-9)*(x-15);

            A:=2; B:=3;   HH:=3500;
        end;

        4:begin

```

```

S:=0.4580*(x-15)*(x-17)+
-2.5000*(x-9)*(x-17)+
2.1875*(x-9)*(x-15);

A:=3; B:=7; HH:=4500;
end;
5:begin
S:=0.5830*(x-15)*(x-17)+
-3.4170*(x-9)*(x-17)+
3.0630*(x-9)*(x-15);

A:=4; B:=10; HH:=5500;
end;
6:begin
S:=0.7500*(x-15)*(x-17)+
-4.5833*(x-9)*(x-17)+
4.1250*(x-9)*(x-15);

A:=6; B:=14; HH:=6500;
end;
7:begin
S:=0.9790*(x-15)*(x-17)+
-6.1660*(x-9)*(x-17)+
5.6250*(x-9)*(x-15);

A:=7; B:=17; HH:=7500;
end;
8:begin
S:=1.2080*(x-15)*(x-17)+
-7.9170*(x-9)*(x-17)+
7.3750*(x-9)*(x-15);

A:=9; B:=40; HH:=8500;
end;
9:begin
S:=1.5420*(x-15)*(x-17)+
-11.000*(x-9)*(x-17)+
10.313*(x-9)*(x-15);

A:=11; B:=60; HH:=9500;

```

```

        end;

10: begin
        S:=11.500*(x-9)+94;
        A:=16; B:=80; HH:=10500;
        end;

end;

S:=S+0.1*TE*(((B-A)/8)*(x-9)+A); {w kg}
if flaga=1 then begin
S1:=S; W:=W+1; HH1:=HH; flaga:=2; goto skok_S;
        end
        else
        begin
        S2:=S; HH2:=HH;
        end;

if H=Hkk then
        begin
        Swzn:=((S2-S1)/(HH2-HH1))*(H-HH1)+S1;
        flaga:=1;
        W:=W-1; H:=Hp;
        goto skok_S;
        end
        else
        begin
        Swzn:=Swzn-((S2-S1)/(HH2-HH1))*(H-HH1)-S1;
        end;
        mst:=mst-Qwzn; Qp:=Qp-Qwzn;
end; {od procedury wznoszenie21b}

begin
end.

```

```

unit dyz_29;
interface
uses
  typy_n, dos, graph, crt;
type
tkom29=array[1..30]of string[30];
tkom=array[1..32] of string[53];
var
tel,ii:longint;
liczba:string[5];
r:char;
Warpopraw:tkom29;
zlawartosc:integer;
Procedure maksymalnyzasieg29(mst,H1,TEM:real;
var Vmkk:real);
Procedure znizanie29(Hkk,Hp:real;
  var Tzn,Qzn,Szn:real);
Procedure oblicz_ck29(mst,H1,vmkk:real;
  woc:integer; var ck:real);
Procedure manewr29(V3,mst,H1,vmkk:real;
  woc:integer; var ck,Tm:real);
Procedure wznoszenie29(gamma, hp, hkk, TE: real;
  moo,woc:integer;
  var Qp,mst: real;
  var Swzn,Twzn,Qwzn:real);

implementation
Procedure maksymalnyzasieg29(mst,H1,TEM:real;
var Vmkk:real);
var
y2,y:real;

begin
  case trunc(H1) of
    0..1000: y2:=0.00005*H1+4;
    1001..2000: y2:=0.00006*(H1-1000)+4.5;
    2001..3000: y2:=0.00008*(H1-2000)+5.1;
    3001..4000: y2:=0.00008*(H1-3000)+5.8;
    4001..5000: y2:=0.00009*(H1-4000)+6.6;

```

```

5001..6000: y2:=0.0001*(H1-5000)+7.4;
6001..7000: y2:=0.00014*(H1-6000)+8.5;
7001..8000: y2:=0.00014*(H1-7000)+9.9;
8001..9000: y2:=0.00018*(H1-8000)+11.3;
9001..10000: y2:=0.00021*(H1-9000)+13.1;
10001..11000: y2:=0.00028*(H1-10000)+15.2;
11001..12000: y2:=0.00030*(H1-11000)+18;
12001..14500: y2:=0.00035*(H1-12000)+21;
end;
y:=0.00005*y2*mst;

Vmkk:=-0.00237*(y-6)*(y-9)*(y-15)
      +0.00772*(y-3)*(y-9)*(y-15)
      -0.00664*(y-3)*(y-6)*(y-15)
      +0.00124*(y-3)*(y-6)*(y-9);
Vmkk:= Vmkk*72.2*sqrt(TEM);{predkosc rz. km/h}
end;
Procedure znizanie29(Hkk, Hp: real; var Tzn, Qzn, Szn: real);
var
x: real;

      BEGIN

      {czas sek}
      x:=0.001*Hkk;
Tzn:=-0.1319*(x-1)*(x-11)+0.1259*(x-1)*(x-6);
      x:=0.001*Hp;
Tzn:=(-0.1319*(x-1)*(x-11)+0.1259*(x-1)*(x-6))-Tzn;
      Tzn:=60*Tzn;
      {paliwo kg}
      x:=0.001*Hkk;
Qzn:=-1.7599*(x-1)*(x-11)+1.5999*(x-1)*(x-6);
      x:=0.001*Hp;
Qzn:=(-1.7599*(x-1)*(x-11)+1.5999*(x-1)*(x-6))-Qzn;

      {droga km}
      x:=0.001*Hkk;
Szn:=-1.3999*(x-1)*(x-11)+1.5199*(x-1)*(x-6);
      x:=0.001*Hp;
Szn:=(-1.3999*(x-1)*(x-11)+1.5199*(x-1)*(x-6))-Szn;
      END;

```

```

Procedure oblicz_ck29(mst,H1,vmkk:real; woc:integer;
                    var ck:real);
var
  m,y1,y2,y,v,okrt:real;
a,b,c,d:real;
begin
  m:=(mst-10000)*0.001;
    case trunc( H1) of
      0..3000:begin
        y1:=0.203*m;
        y2:=0.287*m+1;
        y:=0.00033*(y2-y1)*H1+y1;
          end;
      3001..6000:begin
        y1:=0.287*m+1;
        y2:=0.435*m+2.35;
        y:=0.00033*(y2-y1)*(H1-3000)+y1;
          end;
      6001..8000:begin
        y1:=0.435*m+2.35;
        y2:=0.577*m+3.75;
        y:=0.0005*(y2-y1)*(H1-6000)+y1;
          end;
      8001..10000:begin
        y2:=0.775*m+5.75;
        y1:=0.577*m+3.75;
        y:=0.0005*(y2-y1)*(H1-8000)+y1;
          end;
      10001..12000:begin
        y1:=0.775*m+5.75;
        y2:=1.035*m+8.60;
        y:=0.0005*(y2-y1)*(H1-10000)+y1;
          end;
      12001..15000:begin
        y2:=1.19*m+10.3;
        y1:=1.035*m+8.60;
        y:=0.0005*(y2-y1)*(H1-12000)+y1;
          end;
    end;{od case Hk}
  if y<=20*(Vmkk-0.5) then V:=0.4

```

```

else
begin
case trunc(100*Vmkk) of
50..55:begin
y1:=-0.0033*(y-3)*(y-5)*(y-8)
+0.0667* y *(y-5)*(y-8)
-0.1667* y *(y-3)*(y-8)
+0.1167* y *(y-3)*(y-5);
if y <1 then y2:=0.4
else
y2:=-0.0019*(y-4)*(y-8)*(y-11)
+0.0250*(y-1)*(y-8)*(y-11)
-0.0881*(y-1)*(y-4)*(y-11)
+0.0729*(y-1)*(y-4)*(y-8);
V:=20*(y2-y1)*(Vmkk-0.5)+y1;
end;
56..60:begin
if y<1 then y1:=0.4
else
y1:=-0.0019*(y-4)*(y-8)*(y-11)
+0.0250*(y-1)*(y-8)*(y-11)
-0.0881*(y-1)*(y-4)*(y-11)
+0.0729*(y-1)*(y-4)*(y-8);
if y <2 then y2:=0.4
else
y2:=-0.0022*(y-5)*(y-8)*(y-12)
+0.0302*(y-2)*(y-8)*(y-12)
-0.0694*(y-2)*(y-5)*(y-12)
+0.0450*(y-2)*(y-5)*(y-8);
V:=20*(y2-y1)*(Vmkk-0.55)+y1;
end;
61..65:begin
if y<2 then y1:=0.4
else
y1:=-0.0022*(y-5)*(y-8)*(y-12)
+0.0302*(y-2)*(y-8)*(y-12)
-0.0694*(y-2)*(y-5)*(y-12)
+0.0450*(y-2)*(y-5)*(y-8);
if y <3 then y2:=0.4
else

```

```

        y2:=-0.0017*(y-6)*(y-10)*(y-14)
            +0.0193*(y-3)*(y-10)*(y-14)
            -0.0488*(y-3)*(y-6)*(y-14)
            +0.0313*(y-3)*(y-6)*(y-10);
        V:=20*(y2-y1)*(Vmkk-0.6)+y1;
    end;
66..70:begin
    if y<3 then y1:=0.4
        else
        y1:=-0.0017*(y-6)*(y-10)*(y-14)
            +0.0193*(y-3)*(y-10)*(y-14)
            -0.0488*(y-3)*(y-6)*(y-14)
            +0.0313*(y-3)*(y-6)*(y-10);
        if y <4 then y2:=0.4
            else
            y2:=-0.0033*(y-6)*(y-10)*(y-14)
                +0.0187*(y-4)*(y-10)*(y-14)
                -0.0370*(y-4)*(y-6)*(y-14)
                +0.0234*(y-4)*(y-6)*(y-10);
            V:=20*(y2-y1)*(Vmkk-0.65)+y1;
        end;
71..80:begin
    if y<4 then y1:=0.4
        else
        y1:=-0.0033*(y-6)*(y-10)*(y-14)
            +0.0187*(y-4)*(y-10)*(y-14)
            -0.0370*(y-4)*(y-6)*(y-14)
            +0.0234*(y-4)*(y-6)*(y-10);
        if y<6 then y2:=0.4
            else
            y2:=-0.0050*(y-8)*(y-11)*(y-14)
                +0.0278*(y-6)*(y-11)*(y-14)
                -0.0533*(y-6)*(y-8)*(y-14)
                +0.0285*(y-6)*(y-8)*(y-11);
            V:=10*(y2-y1)*(Vmkk-0.7)+y1;
        end;
81..95:begin
    if y<6 then y1:=0.4
        else
        y1:=-0.0050*(y-8)*(y-11)*(y-14)

```

```

+0.0278*(y-6)*(y-11)*(y-14)
-0.0533*(y-6)*(y-8)*(y-14)
+0.0285*(y-6)*(y-8)*(y-11);
if y<8 then y2:=0.4
else
y2:= 0.00233*(y-11)*(y-14)
-0.1171*(y-8)*(y-14)
+0.1133*(y-8)*(y-11);
V:=10*(y2-y1)*(Vmkk-0.8)+y1;
end;
end; {od trunc(100*vmkk)}
if Vmkk<=0.001*(310-woc)+0.67
then
y:=0.0375*(woc-230)+0.25
else
begin
case woc of
230..269:begin
y1:=-250*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+693*(Vmkk-0.70)*(Vmkk-0.8)*(Vmkk-0.9)
-660*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.9)
+567*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.8);
y2:=-1750*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+5333*(Vmkk-0.70)*(Vmkk-0.8)*(Vmkk-0.9)
-4500*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.9)
+1333*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.8);
y:=0.025*(y2-y1)*(woc-230)+y1;
end;
270..310:begin
y1:=-727.4*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+3333.3*(Vmkk-0.67)*(Vmkk-0.8)*(Vmkk-0.9)
-3461.5*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.9)
+1159.4*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.8);
y2:=-1358.7*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+6166.7*(Vmkk-0.67)*(Vmkk-0.8)*(Vmkk-0.9)
-6615.4*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.9)
+1942.0*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.8);
y:=0.025*(y2-y1)*(woc-270)+y1;
end;
end; {od case woc}

```

```

        end;
    end;
okrt:=5*V+2.5*(y+1)+15;
    case trunc(100*Vmkk) of
50..60:begin
    if okrt<33 then y1:=0.291*(okrt-20)
        else y1:=0.247*(okrt-18);
    if okrt<30 then y2:=0.41*(okrt-18)
        else y2:=0.326*(okrt-15);
        V:=10*(y2-y1)*(Vmkk-0.5)+y1;
        end;
61..70:begin
    if okrt<30 then y1:=0.41*(okrt-18)
        else y1:=0.326*(okrt-15);
    if okrt<27 then y2:=0.549*(okrt-16.5)
        else y2:=0.346*(okrt-10.6);
        V:=10*(y2-y1)*(Vmkk-0.6)+y1;
        end;
71..80:begin
    if okrt<27 then y1:=0.549*(okrt-16.5)
        else y1:=0.346*(okrt-10.6);
    if okrt<24 then y2:=0.940*(okrt-17)
        else y2:=0.392*(okrt-7.6);
        V:=10*(y2-y1)*(Vmkk-0.7)+y1;
        end;
81..95:begin
    if okrt<24 then y1:=0.940*(okrt-17)
        else y1:=0.392*(okrt-7.6);
    if okrt<23 then y2:=2.28*(okrt-19)
        else y2:=0.52*(okrt-5);
        V:=10*(y2-y1)*(Vmkk-0.8)+y1;
        end;
    end; {od case 100*Vmkk}
case trunc(H1) of
    0..1000: y:=-0.0014*H1+12;
    1001..4000: y:=-0.0011*(H1-1000)+10.6;
    4001..6000: y:=-0.0009*(H1-4000)+7.3;
    6001..9000: y:=-0.00063*(H1-6000)+5.5;
    9001..14001: y:=-0.0004*(H1-9000)+3.6;

```

```

end; {od case Hk}
ck:=y*(V+7)/23;
end;
Procedure wznoszenie29(gamma, hp, hkk, TE: real;
moo, woc: integer; var Qp, mst: real;
var Swzn, Twzn, Qwzn: real);
var
tw1, s1, q1, y1, y2,
y, z1, z2, z, s, tw, q, alfa1, alfa2, alfa3, h: real;
m: real;
i: integer;
label
drugiep;
begin
i:=1;
m:=(mst-13000)/1000;
drugiep: case i of
1: h:=trunc(hp);
2: h:=trunc(hkk);
end;
case trunc(h) of
0..200: begin
y1:=0;
y2:=0.05*m+0.8;
y:=((y2-y1)/200)*h;
z2:=0.1*m+0.7;
z:=(z2/200)*h;
end;
201..500: begin
y1:=0.05*m+0.8;
y2:=0.075*m+1.2;
y:=((y2-y1)/300)*(h-200)+y1;
z1:=0.1*m+0.7;
z2:=0.14*m+1.6;
z:=((z2-z1)/300)*(h-200)+z1;
end;
501..1000: begin
y1:=0.075*m+1.2;
y2:=0.12*m+1.9;
y:=((y2-y1)/500)*(h-500)+y1;

```

```

z1:=0.14*m+1.6;
z2:=0.17*m+2.8;
z:=((z2-z1)/500)*(h-500)+z1;
end;
1001..2000:begin
y1:=0.12*m+1.9;
y2:=0.167*m+2.3;
y:=((y2-y1)/1000)*(h-1000)+y1;
z1:=0.17*m+2.8;
z2:=0.24*m+3.2;
z:=((z2-z1)/1000)*(h-1000)+z1;
end;
2001..3000:begin
y1:=0.167*m+2.3;
y2:=0.175*m+2.7;
y:=((y2-y1)/1000)*(h-2000)+y1;
z1:=0.24*m+3.2;
z2:=0.26*m+3.8;
z:=((z2-z1)/1000)*(h-2000)+z1;
end;
3001..4000:begin
y1:=0.175*m+2.7;
y2:=0.24*m+3.0;
y:=((y2-y1)/1000)*(h-3000)+y1;
z1:=0.26*m+3.8;
z2:=0.32*m+4.2;
z:=((z2-z1)/1000)*(h-3000)+z1;
end;
4001..5000:begin
y1:=0.24*m+3.0;
y2:=0.265*m+3.45;
y:=((y2-y1)/1000)*(h-4000)+y1;
z1:=0.32*m+4.2;
z2:=0.34*m+4.7;
z:=((z2-z1)/1000)*(h-4000)+z1;
end;
5001..6000:begin
y1:=0.265*m+3.45;
y2:=0.275*m+4.0;
y:=((y2-y1)/1000)*(h-5000)+y1;

```

```

        z1:=0.34*m+4.7;
        z2:=0.38*m+5.2;
        z:=((z2-z1)/1000)*(h-5000)+z1;
        end;
6001..7000:begin
        y1:=0.275*m+4.0;
        y2:=0.35*m+4.6;
        y:=((y2-y1)/1000)*(h-6000)+y1;
        z1:=0.38*m+5.2;
        z2:=0.46*m+5.6;
        z:=((z2-z1)/1000)*(h-6000)+z1;
        end;
7001..8000:begin
        y1:=0.35*m+4.6;
        y2:=0.48*m+5.5;
        y:=((y2-y1)/1000)*(h-7000)+y1;
        z1:=0.46*m+5.6;
        z2:=0.50*m+6.2;
        z:=((z2-z1)/1000)*(h-7000)+z1;
        end;
8001..9000:begin
        y1:=0.48*m+5.5;
        y2:=0.74*m+6.3;
        y:=((y2-y1)/1000)*(h-8000)+y1;
        z1:=0.50*m+6.2;
        z2:=0.59*m+6.8;
        z:=((z2-z1)/1000)*(h-8000)+z1;
        end;
9001..10000:begin
        y1:=0.74*m+6.3;
        y2:=0.95*m+8.1;
        y:=((y2-y1)/1000)*(h-9000)+y1;
        z1:=0.59*m+6.8;
        z2:=0.71*m+7.4;
        z:=((z2-z1)/1000)*(h-9000)+z1;
        end;
10001..11000:begin
        y1:=0.95*m+8.1;
y2:=1.026*(m-2.5)*(m-3.9)+2.75*(m-2.5)*m-3.71*m*(m-3.9);
        y:=((y2-y1)/1000)*(h-10000)+y1;

```

```

        z1:=0.71*m+7.4;
        z2:=0.86*m+8.1;
        z:=(z2-z1)/1000*(h-10000)+z1;
        end;
end;
{koniec pierwszej charakterystyki}
case woc of
    230..261:begin
y1:=0.119*y*(y-6)*(y-17)-0.167*y*(y-3)*(y-17)
        +0.038*y*(y-3)*(y-6);
y2:=0.1*y*(y-10)*(y-17)-0.2*y*(y-5)
        *(y-17)+0.08*y*(y-5)*(y-10);
        S:=(y2-y1)*(woc-230)/32+y1;
        y1:=0.444*y;
        y2:=0.50*y;
        Tw:=(y2-y1)*(woc-230)/32+y1;
        y1:=46.15*z;
        y2:=51*z;
        Q:=(y2-y1)*(woc-230)/32+y1;
        end;
    262:begin
S:=0.1*y*(y-10)*(y-17)
        -0.2*y*(y-5)*(y-17)+0.08*y*(y-5)*(y-10);
        Tw:=0.50*y;
        Q:=51*z;
        end;
    267:begin
        S:=0.1*y*(y-10)*(y-17)
-0.2*y*(y-5)*(y-17)+0.082*y*(y-5)*(y-10);
        Tw:=0.52*y;
        Q:=505*z;
        end;
    269:begin
        S:=0.1*y*(y-10)*(y-17)
-0.20*y*(y-5)*(y-17)+0.084*y*(y-5)*(y-10);
        Tw:=0.51*y;
        Q:=52.5*z;
        end;
    276:begin
        S:=0.1*y*(y-10)*(y-17)

```

```

-0.21*y*(y-5)*(y-17)+0.086*y*(y-5)*(y-10);
    Tw:=0.514*y;
    Q:=1.59*z*(z-8)*(z-12)
-3.27*z*(z-4)*(z-12)+1.68*z*(z-4)*(z-8);
    end;
284:begin
    S:=0.1*y*(y-10)*(y-17)
-0.22*y*(y-5)*(y-17)+0.095*y*(y-5)*(y-10);
    Tw:=0.517*y;
    Q:=1.61*z*(z-8)*(z-12)
-3.3*z*(z-4)*(z-12)+1.73*z*(z-4)*(z-8);
    end;
289:begin
    S:=0.1*y*(y-10)*(y-17)
-0.23*y*(y-5)*(y-17)+0.1*y*(y-5)*(y-10);
    Tw:=0.52*y;
    Q:=1.62*z*(z-8)*(z-12)
-3.31*z*(z-4)*(z-12)+1.77*z*(z-4)*(z-8);
    end;
291:begin
    S:=0.1*y*(y-10)*(y-17)
-0.23*y*(y-5)*(y-17)+0.112*y*(y-5)*(y-10);
    Tw:=0.522*y;
    Q:=1.625*z*(z-8)*(z-12)
-3.328*z*(z-4)*(z-12)+1.78*z*(z-4)*(z-8);
    end;
298:begin
    S:=0.103*y*(y-10)*(y-17)
-0.234*y*(y-5)*(y-17)+0.126*y*(y-5)*(y-10);
    Tw:=0.525*y;
    Q:=1.656*z*(z-8)*(z-12)
-3.39*z*(z-4)*(z-12)+1.9*z*(z-4)*(z-8);
    end;
306:begin
    S:=0.107*y*(y-10)*(y-17)
-0.237*y*(y-5)*(y-17)+0.140*y*(y-5)*(y-10);
    Tw:=0.012*y*(y-8)*(y-13)
-0.0225*y*(y-3)*(y-13)+0.0123*y*(y-3)*(y-8);
    Q:=1.679*z*(z-8)*(z-12)
-3.44*z*(z-4)*(z-12)+1.95*z*(z-4)*(z-8);

```

```

        end;
    end;
    (* Uwzględnienie temperatury *)
    s:=s/10;
    alfa1:=0.01745*(0.0255*s*(s-8.6)*(s-17)
    -0.0321*s*(s-3)*(s-17)+0.013*s*(s-3)*(s-8.6));
    s:=s*10;
    alfa2:=0.01745*(0.0255*tw*(tw-8.6)*(tw-17)
    -0.0321*tw*(tw-3)*(tw-17)+0.013*tw*(tw-3)*(tw-8.6));
    Q:=Q/100;
    alfa3:=0.01745*(0.326*Q*(Q-3.5)*(Q-7.75)
    -0.384*Q*(Q-1.75)*(Q-7.75)+0.132*Q*(Q-1.75)*(Q-3.5));
    Q:=Q*100;
    Q:=trunc(10*(sin(alfa3)/cos(alfa3))*TE+Q);
    S:=4*(sin(alfa1)/cos(alfa1))*TE+S;
    Tw:=0.2*(sin(alfa2)/cos(alfa2))*TE+Tw;
    case i of
        1:begin
            Q1:=Q;
            S1:=S;
            Tw1:=Tw;
            i:=i+1;
            goto drugiep;
        end;
        2:begin
            mst:=mst-(Q-Q1);
            Qp:=Qp-(Q-Q1); Qwzn:=Q-Q1;
            Swzn:=trunc(S-S1);
            Twzn:=trunc((Tw-Tw1)*60)    {w sek};
            end;
    end;
end;
end;

Procedure manewr29(V3,mst,H1,vmkk:real; woc:integer;
    var ck,Tm:real);
var
m,y1,y2,y,v,okrt,a,b,c,d:real;
flaga,W:integer;
Nzo,Nzo1,Nzo2,x:real;

```

```

label
skok_N, skok_Mst;
BEGIN
    W:=trunc(0.001*H1)+1;
    flaga:=1;
    skok_N: case W of
        1:Nzo:=1.1;
        2:Nzo:=1.25;
        3:Nzo:=1.4;
        4:Nzo:=1.6;
        5:Nzo:=1.8;
        6:Nzo:=2.05;
        7:Nzo:=2.37;
        8:Nzo:=2.7;
        9:Nzo:=3.12;
        10:Nzo:=3.5;
        11:Nzo:=4.05;
    end;
    if flaga=1 then begin
        Nzo1:=Nzo; flaga:=2; W:=W+1;
        goto skok_N;
    end
    else
        begin
            Nzo2:=Nzo;
        end;
    Nzo:=0.001*(Nzo2-Nzo1)*(H1-1000*(W-1))+Nzo1;
    W:=trunc(0.001*Mst);
    flaga:=1;

    skok_Mst: case W of
        12..13: y:=2.3*(Nzo-1)+0.8;
        14..15: y:=2.75*(Nzo-1)+1;
        16..17: y:=3.125*(Nzo-1)+1.5;
        18..19: y:=3.5*(Nzo-1)+2;
        20..25: y:=3.9*(Nzo-1)+2.4;
    end;
    if flaga=1 then
        begin
            y1:=y; W:=W+2; flaga:=0; goto skok_Mst;

```

```

        end
    else
        begin
            y2:=y;
        end;

y:=(y2-y1)/2000*(Mst-1000*(W-2))+y1;

if y<=20*(Vmkk-0.5) then V:=0.4
    else
        begin

case trunc(100*Vmkk) of
    50..55:begin
        y1:=-0.0033*(y-3)*(y-5)*(y-8)
            +0.0667* y *(y-5)*(y-8)
            -0.1667* y *(y-3)*(y-8)
            +0.1167* y *(y-3)*(y-5);
        if y <1 then y2:=0.4
            else
                y2:=-0.0019*(y-4)*(y-8)*(y-11)
                    +0.0250*(y-1)*(y-8)*(y-11)
                    -0.0881*(y-1)*(y-4)*(y-11)
                    +0.0729*(y-1)*(y-4)*(y-8);
                V:=20*(y2-y1)*(Vmkk-0.5)+y1;
            end;
        56..60:begin
            if y<1 then y1:=0.4
                else
                    y1:=-0.0019*(y-4)*(y-8)*(y-11)
                        +0.0250*(y-1)*(y-8)*(y-11)
                        -0.0881*(y-1)*(y-4)*(y-11)
                        +0.0729*(y-1)*(y-4)*(y-8);
                    if y <2 then y2:=0.4
                        else
                            y2:=-0.0022*(y-5)*(y-8)*(y-12)
                                +0.0302*(y-2)*(y-8)*(y-12)
                                -0.0694*(y-2)*(y-5)*(y-12)
                                +0.0450*(y-2)*(y-5)*(y-8);

```

```

V:=20*(y2-y1)*(Vmkk-0.55)+y1;

        end;
61..65:begin
if y<2 then y1:=0.4
        else
y1:=-0.0022*(y-5)*(y-8)*(y-12)
      +0.0302*(y-2)*(y-8)*(y-12)
      -0.0694*(y-2)*(y-5)*(y-12)
      +0.0450*(y-2)*(y-5)*(y-8);

if y <3 then y2:=0.4
        else
y2:=-0.0017*(y-6)*(y-10)*(y-14)
      +0.0193*(y-3)*(y-10)*(y-14)
      -0.0488*(y-3)*(y-6)*(y-14)
      +0.0313*(y-3)*(y-6)*(y-10);

V:=20*(y2-y1)*(Vmkk-0.6)+y1;
        end;
66..70:begin
if y<3 then y1:=0.4
        else
y1:=-0.0017*(y-6)*(y-10)*(y-14)
      +0.0193*(y-3)*(y-10)*(y-14)
      -0.0488*(y-3)*(y-6)*(y-14)
      +0.0313*(y-3)*(y-6)*(y-10);

if y <4 then y2:=0.4
        else
y2:=-0.0033*(y-6)*(y-10)*(y-14)
      +0.0187*(y-4)*(y-10)*(y-14)
      -0.0370*(y-4)*(y-6)*(y-14)
      +0.0234*(y-4)*(y-6)*(y-10);

V:=20*(y2-y1)*(Vmkk-0.65)+y1;

        end;
71..80:begin

```

```

if y<4 then y1:=0.4
    else
y1:=-0.0033*(y-6)*(y-10)*(y-14)
    +0.0187*(y-4)*(y-10)*(y-14)
    -0.0370*(y-4)*(y-6)*(y-14)
    +0.0234*(y-4)*(y-6)*(y-10);
if y<6 then y2:=0.4
    else
y2:=-0.0050*(y-8)*(y-11)*(y-14)
    +0.0278*(y-6)*(y-11)*(y-14)
    -0.0533*(y-6)*(y-8)*(y-14)
    +0.0285*(y-6)*(y-8)*(y-11);

V:=10*(y2-y1)*(Vmkk-0.7)+y1;

    end;
81..95:begin
if y<6 then y1:=0.4
    else
y1:=-0.0050*(y-8)*(y-11)*(y-14)
    +0.0278*(y-6)*(y-11)*(y-14)
    -0.0533*(y-6)*(y-8)*(y-14)
    +0.0285*(y-6)*(y-8)*(y-11);
if y<8 then y2:=0.4
    else
y2:= 0.00233*(y-11)*(y-14)
    -0.1171*(y-8)*(y-14)
    +0.1133*(y-8)*(y-11);
V:=10*(y2-y1)*(Vmkk-0.8)+y1;
    end;

end;{od trunc(100*vmkk)}
if Vmkk<=0.001*(310-woc)+0.67
then y:=0.0375*(woc-230)+0.25
else
begin
case woc of
230..269:begin
y1:=-250*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)

```

```

+693*(Vmkk-0.70)*(Vmkk-0.8)*(Vmkk-0.9)
-660*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.9)
+567*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.8);

y2:=-1750*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+5333*(Vmkk-0.70)*(Vmkk-0.8)*(Vmkk-0.9)
-4500*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.9)
+1333*(Vmkk-0.70)*(Vmkk-0.75)*(Vmkk-0.8);
y:=0.025*(y2-y1)*(woc-230)+y1;

end;
270..310:begin
y1:=-727.4*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+3333.3*(Vmkk-0.67)*(Vmkk-0.8)*(Vmkk-0.9)
-3461.5*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.9)
+1159.4*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.8);
y2:=-1358.7*(Vmkk-0.75)*(Vmkk-0.8)*(Vmkk-0.9)
+6166.7*(Vmkk-0.67)*(Vmkk-0.8)*(Vmkk-0.9)
-6615.4*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.9)
+1942.0*(Vmkk-0.67)*(Vmkk-0.75)*(Vmkk-0.8);
y:=0.025*(y2-y1)*(woc-270)+y1;

end;
end;{od case woc}
end;
end;

okrt:=5*V+2.5*(y+1)+15;
case trunc(100*Vmkk) of
50..60:begin
if okrt<33 then y1:=0.291*(okrt-20)
else y1:=0.247*(okrt-18);
if okrt<30 then y2:=0.41*(okrt-18)
else y2:=0.326*(okrt-15);
V:=10*(y2-y1)*(Vmkk-0.5)+y1;
end;
61..70:begin
if okrt<30 then y1:=0.41*(okrt-18)
else y1:=0.326*(okrt-15);
if okrt<27 then y2:=0.549*(okrt-16.5)
else y2:=0.346*(okrt-10.6);

```

```

V:=10*(y2-y1)*(Vmkk-0.6)+y1;

end;

71..80:begin
if okrt<27 then y1:=0.549*(okrt-16.5)
else y1:=0.346*(okrt-10.6);
if okrt<24 then y2:=0.940*(okrt-17)
else y2:=0.392*(okrt-7.6);
V:=10*(y2-y1)*(Vmkk-0.7)+y1;

end;

81..95:begin
if okrt<24 then y1:=0.940*(okrt-17)
else y1:=0.392*(okrt-7.6);
if okrt<23 then y2:=2.28*(okrt-19)
else y2:=0.52*(okrt-5);
V:=10*(y2-y1)*(Vmkk-0.8)+y1;

end;

end; {od case 100*Vmkk}

case trunc(H1) of
0..1000: y:=-0.0014*H1+12;
1001..4000: y:=-0.0011*(H1-1000)+10.6;
4001..6000: y:=-0.0009*(H1-4000)+7.3;
6001..9000: y:=-0.00063*(H1-6000)+5.5;
9001..14001: y:=-0.0004*(H1-9000)+3.6;

end;{od case Hk}

ck:=y*(V+7)/23;

for i:=1 to 8 do Tm:=0.308*V3;

End;begin
end. sound(500+180*i);
delay(3);
nosound;
sound(1800-180*i);
delay(1);
nosound;
end;

```

```

unit pocz;
interface
uses
  typy_n, dos, graph, crt;
var
  r: char;
type
  tkom29=array[1..30]of string[30];
  tkom60=array[1..3] of string[100];
const
  Tabpocz:tkom60=
    (' - czasu dyzuruwania
    pojedynczego samolotu ',
    ' - mozliwosci ciaglego dyzuruwania
    zadanej liczby samolotow ',
    ' - potrzebnej liczby samolotow do
    dyzuruwania w zadanych warunkach');
  procedure muzyka;
  procedure scroll60(il,px,py,kx,ky: integer;
    obiekt:tkom60; var k: integer);
  Procedure pocz1(var iltabpocz: integer);
implementation
  procedure muzyka;
  var
    k,j,i: integer;
  begin
    for k:=1 to 3 do begin
      for j:=1 to 25 do
        begin
          for i:=1 to 5 do
            begin
              sound(500+180*i);
              delay(3);
              nosound;
              sound(1500-180*i);
              delay(1);
              nosound;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

delay(5);
end;
    delay(15);
        end;
end;

procedure scroll60(il,px,py,kx,ky:integer;
    obiekt:tkom60; var k:integer);
var
i,m,n,z : integer;
begin
    SetFillStyle(0,0);
    k:=1;
    m:=1;
    if il<18 then n:=il
        else n:=18;
        repeat
    for i:=m to n do
        begin
            bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
            if i=k then
                begin
                    setcolor(0);
                    SetFillStyle(1,1);
                    setcolor(0);
                    bar(px,py+(i-m)*15,kx,ky+(i-m)*15);
                    SetFillStyle(0,0);
                    outtextxy(px+10,py+4+(i-m)*15,obiekt[i]);
                    SetColor(1);
                end
            else
                outtextXY(px+4,py+4+(i-m)*15,obiekt[i]);
            end;
        end;
    OuttextXY(10,333,'WYBIERZ WARIANT I NACISNIJ "Enter" ');
    repeat
        r:=readkey;
        if r=#0 then
            begin

```

```

r:=readkey;
z:=0;
case r of
'I' : begin
      k:=1;
      m:=1;
      if il<18 then n:=il
      else n:=18;
      z:=1;
    end;
'Q' : begin
      k:=il;
      if il<18 then m:=1
      else m:=il-17;
      n:=il;
      z:=1;
    end;
'H' : begin
      k:=k-1;
      if k=0 then
        begin
          k:=1;
          Sound(800);
          Delay(300);
          Nosound;
        end
      else
        begin
          z:=1;
          if k<m then begin
            m:=m-1;
            n:=n-1;
          end
          else
            begin
              z:=0;
              SetFillStyle(1,1);
              bar(px,py+(k-m)*15,kx,ky+(k-m)*15);
              SetFillStyle(0,0);
              SetColor(0);
            end;
          end;
        end;

```

```

        outtextXY(px+4, py+4+(k-m)*15, obiekt[k]);
        SetColor(1);
bar(px, py+(k-m+1)*15, kx, ky+(k-m+1)*15);
        outtextXY(px+4, py+4+(k-m+1)*15, obiekt[k+1]);
                end;
        end;
end;
'P' : begin
        k:=k+1;
        if k>il then
                begin
                        k:=il;
                        Sound(800);
                        Delay(300);
                        Nosound;
                end
        else
                begin
                        z:=1;
                        if k>n then begin
                                n:=n+1;
                                m:=m+1;
                        end
                        else
                                begin
                                        z:=0;
                                        SetFillStyle(1,1);
                                        bar(px, py+(k-m)*15, kx, ky+(k-m)*15);
                                        SetFillStyle(0,0);
                                        SetColor(0);
                                        outtextXY(px+4, py+4+(k-m)*15, obiekt[k]);
                                        SetColor(1);
                                        bar(px, py+(k-m-1)*15, kx, ky+(k-m-1)*15);
                                        outtextXY(px+4, py+4+(k-m-1)*15, obiekt[k-1]);
                                                end;
                                end;
                end;
        end;
end;
else

```

```

        if r<>#13 then
            begin
                Sound(800);
                Delay(300);
                Nosound;
            end;
        until (r=#13)or(z=1)
    until r=#13;
    SetFillStyle(1,1);
end;
Procedure pocz1(var iltabpocz:integer);
begin
setcolor(0);
bar(10,10,120,30);
outtextxy(20,20,'OKRESLANIE:');
setcolor(15);
rectangle(1,1,getmaxx,getmaxy);
rectangle(28,58,602,110);
scroll160(3,30,60,600,77,Tabpocz,iltabpocz);
clearviewport;
end;
end.

```

```

unit typy_n;
interface
uses
dos, graph, crt;
type
tkom29=array[1..30]of string[30];
tkom=array[1..32] of string[53];
var
flag, V1, V2, V3, TEL, H1, Qw, H22, S1, H11,
S3, H33: integer;
Tdyz1, Td, Ta1, Ta3, Vmp, Mst1, Qp1, Mstwe,
Qpwe, y, Qz, Qmax, Hp, Hkk, Qzn, Szn, Tzn: real;
Tm, TEM, TE, TEM1, TEM2, TEM3, mst, H1, V1P,
V1M, V2P, V2M, V3P, V3M, Vmkk,
mz: real;
Mstu, Wocu, iltzas, iltzb, iluz, V1mz,
V2mz, V3mz: integer;
Wocz, Tlad, Qlad, mr, wocr, k, woc: integer;
zmienne: array[1..20] of string[30];
iltabpocz: integer;
ck: real;
Qp, gamma,
Swzn, Twzn, Qwzn, Vp, Czasna_z: real;

moo, zb: integer;
const
TMst: array[1..30, 1..10] of real=
((12453, 284, 4200, 0, -1, 38, 60, 1, 506, 17),
(12569, 306, 5700, 0, -1, 38, 60, 0, 506, 17),
(12161, 276, 4200, 0, -1, 38, 60, 1, 506, 17),
(12277, 298, 5700, 0, -1, 38, 60, 0, 506, 17),
(12453, 276, 4200, 0, -1, 21, 23, 1, 209, 12),
(12570, 298, 5700, 0, -1, 21, 23, 0, 209, 12),
(11715, 262, 4200, 1, 1, 12, 23, 1, 90, 8),
(11831, 284, 5700, 1, 1, 12, 23, 0, 90, 8),
(12007, 269, 4200, 0, -1, 21, 23, 1, 209, 12),
(12123, 291, 5700, 0, -1, 21, 23, 0, 209, 12),
(11861, 267, 4200, 0, -1, 21, 23, 1, 209, 12),

```

(11977, 289, 5700, 0, -1, 21, 23, 0, 209, 12),  
 (12127, 269, 4200, 0, -1, 38, 60, 1, 506, 17),  
 (12243, 291, 5700, 0, -1, 38, 60, 0, 506, 17),  
 (11981, 267, 4200, 0, -1, 38, 60, 1, 506, 17),  
 (12097, 289, 5700, 0, -1, 38, 60, 0, 506, 17),  
 (11681, 256, 4200, 0, -1, 21, 23, 1, 209, 12),  
 (11797, 276, 5700, 0, -1, 21, 23, 0, 209, 12),  
 (11827, 262, 4200, 0, -1, 21, 23, 1, 209, 12),  
 (11943, 284, 5700, 0, -1, 21, 23, 0, 209, 12),  
 (11535, 252, 4200, 1, 1, 12, 23, 1, 90, 8),  
 (11651, 276, 5700, 1, 1, 12, 23, 0, 90, 8),  
 (11801, 254, 4200, 0, -1, 38, 60, 1, 506, 17),  
 (11917, 276, 5700, 0, -1, 38, 60, 0, 506, 17),  
 (11501, 245, 4200, 0, -1, 21, 23, 1, 209, 12),  
 (11617, 267, 5700, 0, -1, 21, 23, 0, 209, 12),  
 (11355, 241, 4200, 1, 1, 12, 23, 1, 90, 8),  
 (11471, 262, 5700, 1, 1, 12, 23, 0, 90, 8),  
 (11295, 237, 4200, 0, -1, 38, 60, 1, 0, 0),  
 (11411, 259, 5700, 0, -1, 38, 60, 0, 0, 0));

Twydruk:array[1..16] of string[50]=

('Typ myśliwca ..... ',  
 'Wariant uzbrojenia ..... ',  
 'Gęstość paliwa ..... ',  
 'Czas pracy na ziemi [min.]..... ',  
 'Zapasy nawigacyjny..... ',  
 'Średnia temp. w rejonie lotów .... [stop. C ]... ',  
 'Średnia wysokość rejonu lotów ..... [metry]... ',  
 'Paliwo walki powietrznej..... [kg]..... ',  
 'Odległość strefy dyżurowania ..... [km]..... ',  
 'Wysokość dolotu do strefy..... [m]..... ',  
 'Prędkość dolotu do strefy..... [km/h].... ',  
 'Wysokość dyżurowania..... [m]..... ',  
 'Prędkość dyżurowania..... [km/h].... ',  
 'Odległość do lotniska lądowania..... [km ]..... ',  
 'Wysokość powrotu..... [m]..... ',  
 'Prędkość powrotu..... [km/h].... ');

implementation

end.

