

Grey Scale #13



DANES-PICTA.COM

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

AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

SYSTEM MODELOWANIA WALKI ZBROJNEJ

"MODEL-1"

KOMPUTEROWA MAPA WSPÓLCZYNNIKÓW

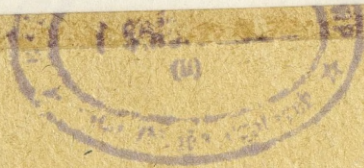
"ŚRODOWISKO"

59692

Biblioteka Główna
Akademii Obrony Narodowej
571128



00-001441-001-0



WARSZAWA

1990



AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

SYSTEM MODELOWANIA WALKI ZBROJNEJ

"MODEL-1"

KOMPUTEROWA MAPA WSPÓŁCZYNNIKÓW

"ŚRODOWISKO"

1
59692

Biblioteka Główna
Akademii Obrony Narodowej

~~571128~~

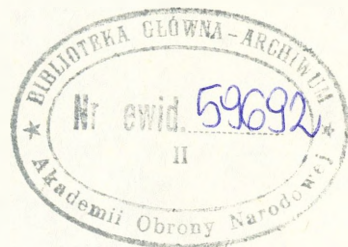


~~01-001441-001-0~~

WARSZAWA

1990

1



SYSTEM MODELOWANIA WALKI ZBROJNEJ "MODEL - 1"

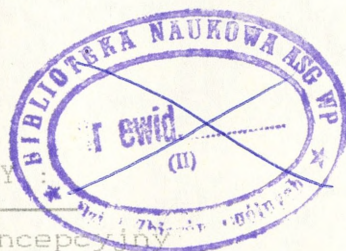
KOMPUTEROWA MAPA WSPÓŁCZYNNIKÓW

"ŚRODOWISKO"

1992

1

~~5/1128~~



AUTORZY

projekt koncepcyjny
oraz założenia
oprogramowanie

płk dr J. Stępak
ppłk mgr inż S. Dąbrowski

WSTĘP

Ocena wpływu elementów rzeźby i pokrycia terenu, infrastruktury technicznej, operacyjnego przygotowania TDW, warunków klimatycznych, rozbudowy inżynieryjnej terenu na warunki działań bojowych, prowadzona metodami tradycyjnymi, nie jest możliwa do zastosowania w badaniach symulacyjnych i grach wojennych prowadzonych przy pomocy komputerowej techniki obliczeniowej. Dlatego też zrodził się problem opracowania metody oceny wpływu wymienionych elementów na efektywność walki zbrojnej, możliwej do zastosowania w symulacjach i grach komputerowych. Jako jedno z rozwiązań przyjęto opracowanie komputerowej mapy współczynników.

Istota rozwiązania problemu polega na przyjęciu stosowanego systemu podziału map na arkusze, który jest zsynchronizowany z podziałem arkuszowym Międzynarodowej Mapy Świata w skali 1 : 1000000. Dla każdego arkusza mapy, według określonej metodyki obliczane są współczynniki wpływu elementów rzeźby i pokrycia terenu, infrastruktury technicznej, operacyjnego przygotowania TDW, rozbudowy inżynieryjnej terenu na warunki prowadzenia działań bojowych. W ten sposób powstaje mapa współczynników. Ze względu jednak na to, że problem obliczania współczynników, ewidencjonowania, uaktualniania jest bardzo pracochłonny, dlatego wykonanie tych czynności powierzono komputerowi. Tak powstała koncepcja opracowania komputerowej mapy współczynników.

Rozwiązanie tego problemu rozłożono na cztery etapy:

- w etapie pierwszym opracowano program ewidencjonowania, przechowywania i uaktualniania, obliczonych "sposobem ręcznym", współczynników;
- w drugim etapie przewiduje się opracowanie programu zapewniającego obliczanie współczynników przez komputer, na podstawie założonych algorytmów, z wprowadzonych do komputera danych liczbowych, przygotowanych w dalszym ciągu "sposobem ręcznym";
- w trzecim etapie przewiduje się automatyczne obliczanie współczynników uśrednionych dla określonych obszarów terenu i wprowadzanie ich do poszczególnych kroków prowadzonej symulacji /gry/ komputerowej;
- w czwartym i ostatnim etapie przewiduje się automatyczne zbieranie danych z map topograficznych /specjalnych/ lub innych dokumentów, obliczanie współczynników, uśrednianie ich dla badanego obszaru terenu i automatyczne wprowadzanie do poszczególnych kroków symulacji lub gry komputerowej.

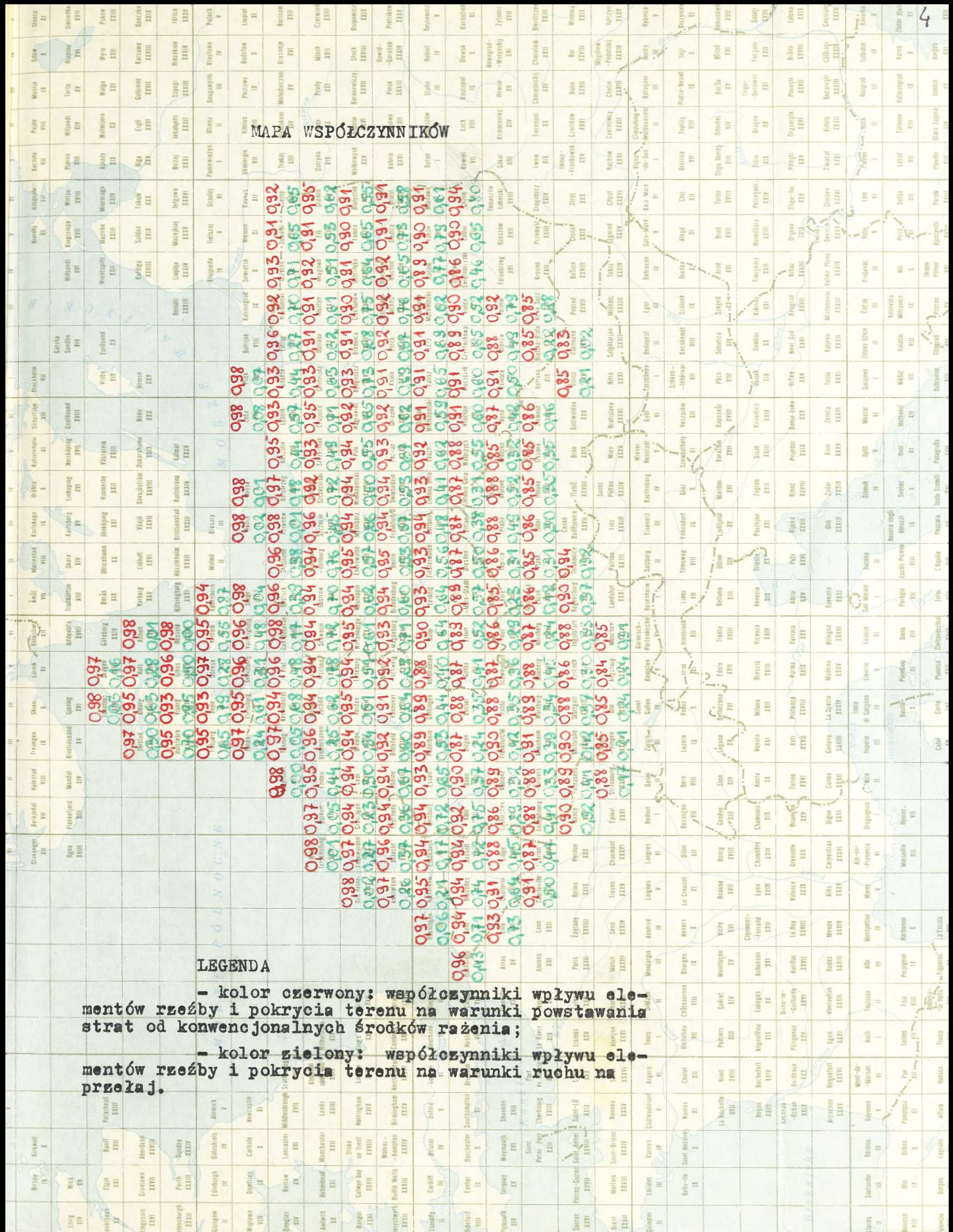
Niniejsze opracowanie prezentuje pierwszy etap rozwiązania problemu.

W ramach systemu modelowania walki zbrojnej "MODEL - 1" obliczono współczynniki wpływu elementów rzeźby i pokrycia terenu na warunki powstawania strat od konwencjonalnych środków rażenia i wpływu elementów rzeźby i pokrycia terenu na warunki ruchu na przełaj. Jako podstawę do obliczeń przyjęto arkusze mapy w skali 1 : 200 000. Współczynniki te zobrazowane są na mapie współczynników. Przy pomocy prezentowanego programu komputerowego, współczynniki te zostały wprowadzone do pamięci komputera i tam są przechowywane. Na żądanie mogą one być przedstawione w formie tabelarycznego wydruku. Opracowany program umożliwia przechowywanie i uaktualnianie oraz sporządzanie zestawień po 200 czynników liczbowych dla każdego arkusza mapy w skalach od 1 : 25000 do 1 : 1000000.

MAPA WSPÓŁCZYNNIKÓW

LEGENDA

- kolor czerwony: współczynniki wpływu elementów rzeźby i pokrycia terenu na warunki powstawania strat od konwencjonalnych środków rażenia;
- kolor zielony: współczynniki wpływu elementów rzeźby i pokrycia terenu na warunki ruchu na przelaj.



Program "Wspomagamanie" służy do tworzenia i aktualizacji bazy danych o terenie. Teren charakteryzowany jest poprzez współczynniki, a współczynnik może być podany jako wcześniej określona liczba z zakresu 0.0 - 1.0 lub jako wektor elementów określających dany teren z którego współczynnik jest obliczany w/g wzoru:

$$W_j = \frac{\sum_{i=1}^{10} w_i}{i} \quad \text{dla } j = 1, 20$$

W zależności od potrzeb dane opisują teren obrazowany mapą odpowiedniej skali. W programie możliwe jest tworzenie bazy danych dla skal map od 1:1000000 do 1:25000. Dane przechowywane są w zbiorach o nazwach zgodnych z godłem mapy w odpowiedniej skali.

Program realizuje następujące funkcje :

1. Wprowadzanie danych - podczas wprowadzania danych na ekranie widoczna jest mapa terenu który opisujemy.
2. Aktualizacja danych - wyświetlana jest mapa oraz wcześniej wprowadzone dane.
3. Wydruk danych - możliwe jest oglądanie wprowadzanych tabel danych na ekranie.

3. TECHNOLOGIA OBSŁUGI PROGRAMU

Program wywołujemy komendą

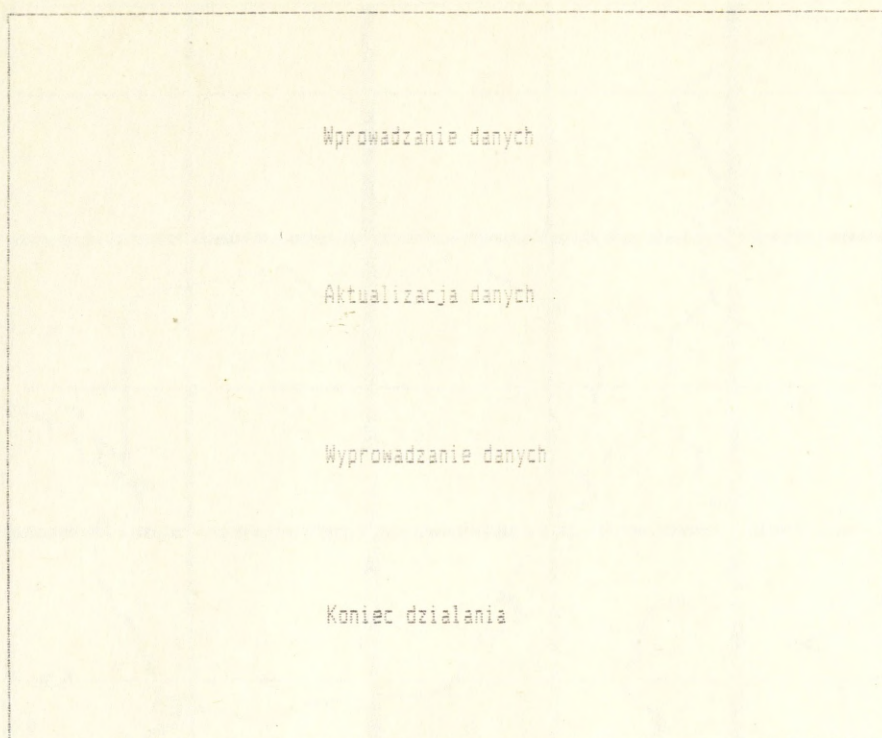
WSPOMAGA

Po rozpoczęciu wykonywania programu wyświetlana jest
wineta programu

AKADEMIA OBRONY NARODOWEJ		
Komputerowa mapa współczynników		
"S R O D O W I S K O"		
AUTORZY :		
projekt koncepcyjny oraz złożenia	pplk dr J. Stepak	
oprogramowanie	pplk mgr inż S. Dabrowski	
WARSZAWA	GRUDZIEŃ	1989

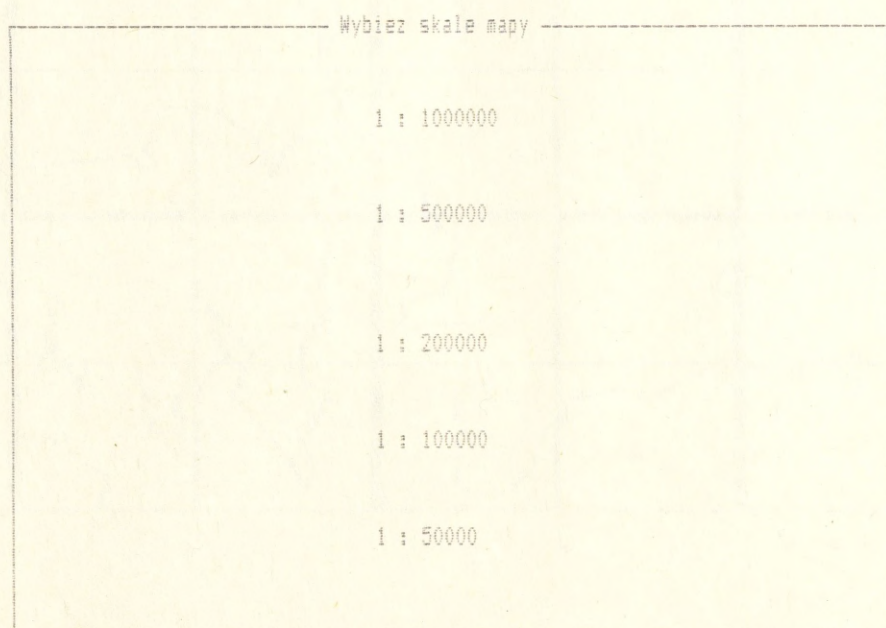
przejdzie do dalszej realizacji programu następuje po
naciśnięciu dowolnego klawisza i wyświetlane są

S R O D O W I S K O
W P R O W A D Z E N I E.
A K T U A L I Z A C J A
O B L I C Z A N I E

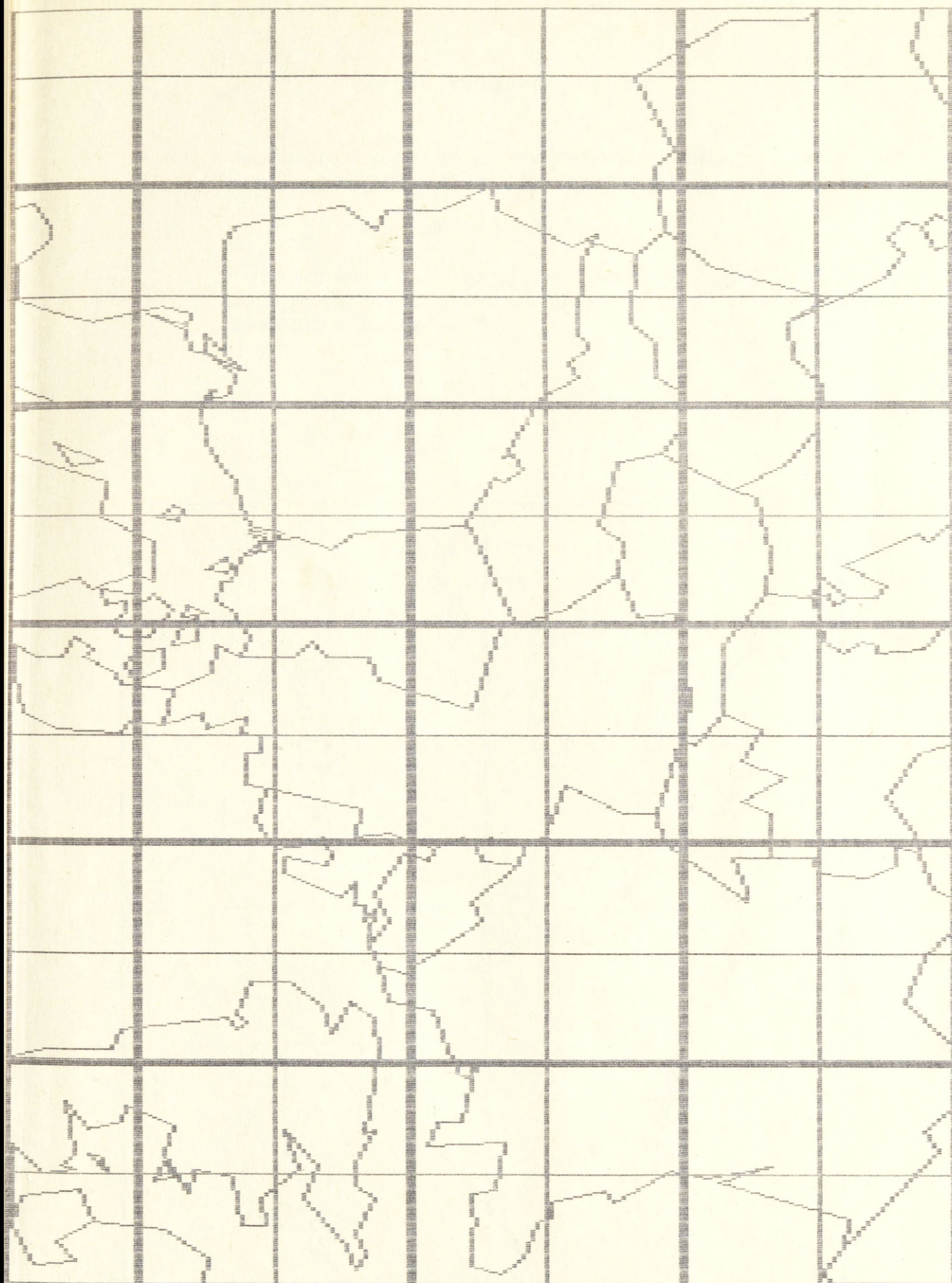


po przesunięciu za pomocą kursora na czynność którą chcemy realizować i wciśnięciu Enter/wybranie czynności/ przechodzimy do realizacji odpowiednich funkcji programu :

1. Wprowadzanie danych



wybieramy skalę mapy na której będziemy wprowadzali dane /za pomocą kursora i klawisza Enter/.



Mapa jest podzielona za pomocą linii na mapy w wybranej skali/do skali 1:200000/, naciśnięcie dowolnego klawisza powoduje przejści do realizacji funkcji i pojawia się na ekranie

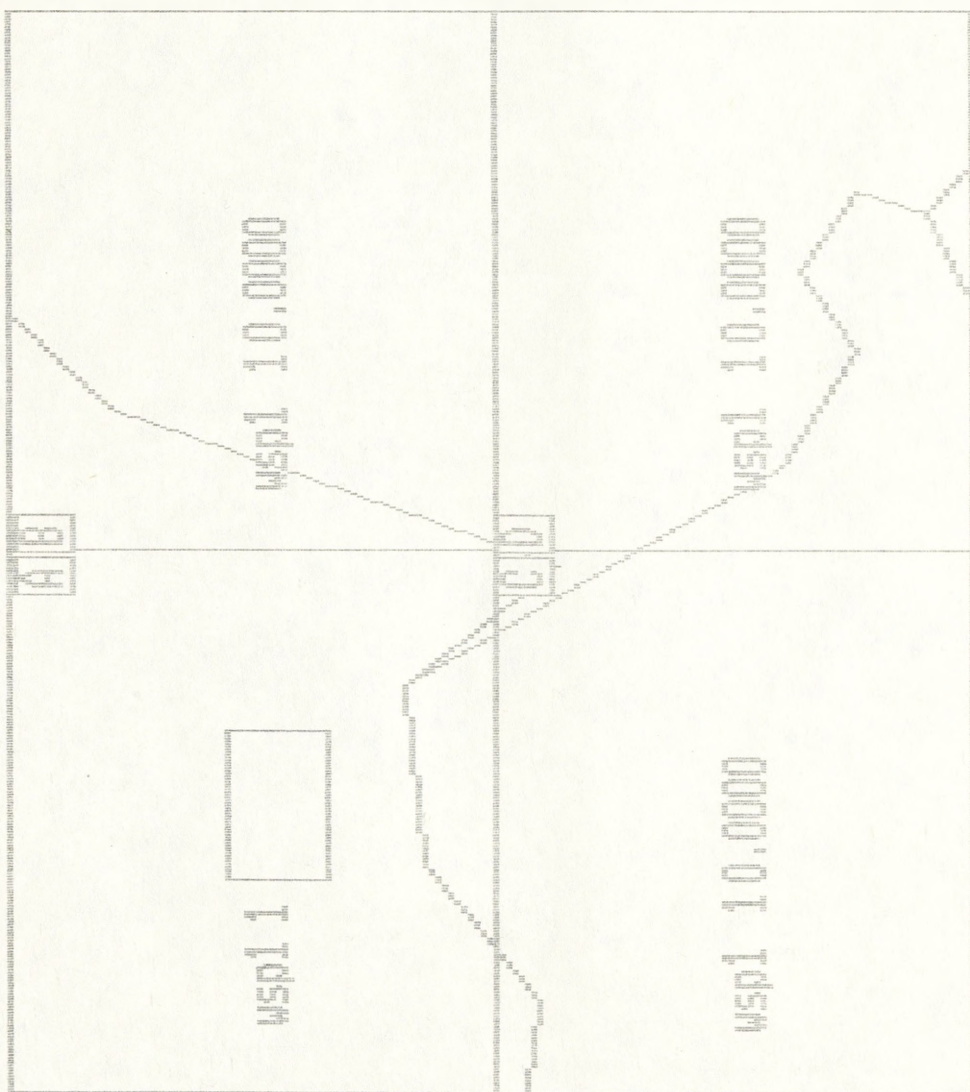
Nazwa kwadratu mapy

następnie należy wprowadzić godło mapy od której zaczynamy wprowadzanie danych. Np

N 31 A 1 1 A

Wprowadzanie każdego elementu godła kończy naciśnięcie klawisza Enter, a po ostatnim elemencie klawisz End.

L - 34



wsp1

1

0.000

nazwa współcz. nr. elementu wektora wartość pocz.

Enter - koniec pola END - koniec wektora skład. ESC - koniec współczynnika

Ramka wskazuje miejsce aktualnie wprowadzanych danych, a liczby wprowadzane mogą być poprzedzone prowadzącym zerem lub nie/.00 lub 0.0/.

Przy wprowadzaniu mogą zaistnieć następujące przypadki:

1. gdy współczynnik został wcześniej określony liczbą poprzedzona jest literą "w"/nie jest wyświetlana/
2. gdy długość wektora składowego jest mniejsza od 10 po ostatnim elemencie naciskamy klawisz End
3. gdy mapę opisuje mniej niż 20 współczynników to po danych składowych naciskamy Esc.

Po zakończeniu wprowadzania danych określających współczynnik następuje obliczenie wartości współczynnika i wartość zostaje wyświetlona na ekranie.

W ostatnim wierszu ekranu pojawia się pytanie :

Czy wartość jest prawidłowa?

opcjonalnie przyjmowana jest odpowiedź TAK naciśnięcie klawisza Enter powoduje przyjęcie jej. W przeciwnym wypadku należy wprowadzić odpowiedź.

W zależności od podanej odpowiedzi są możliwe następujące działania:

NIE - wektor elementarny przeglądany jest od początku i błędny element wektora należy wprowadzić jeszcze raz, a prawidłowo wprowadzone zatwierdzić klawiszem Enter

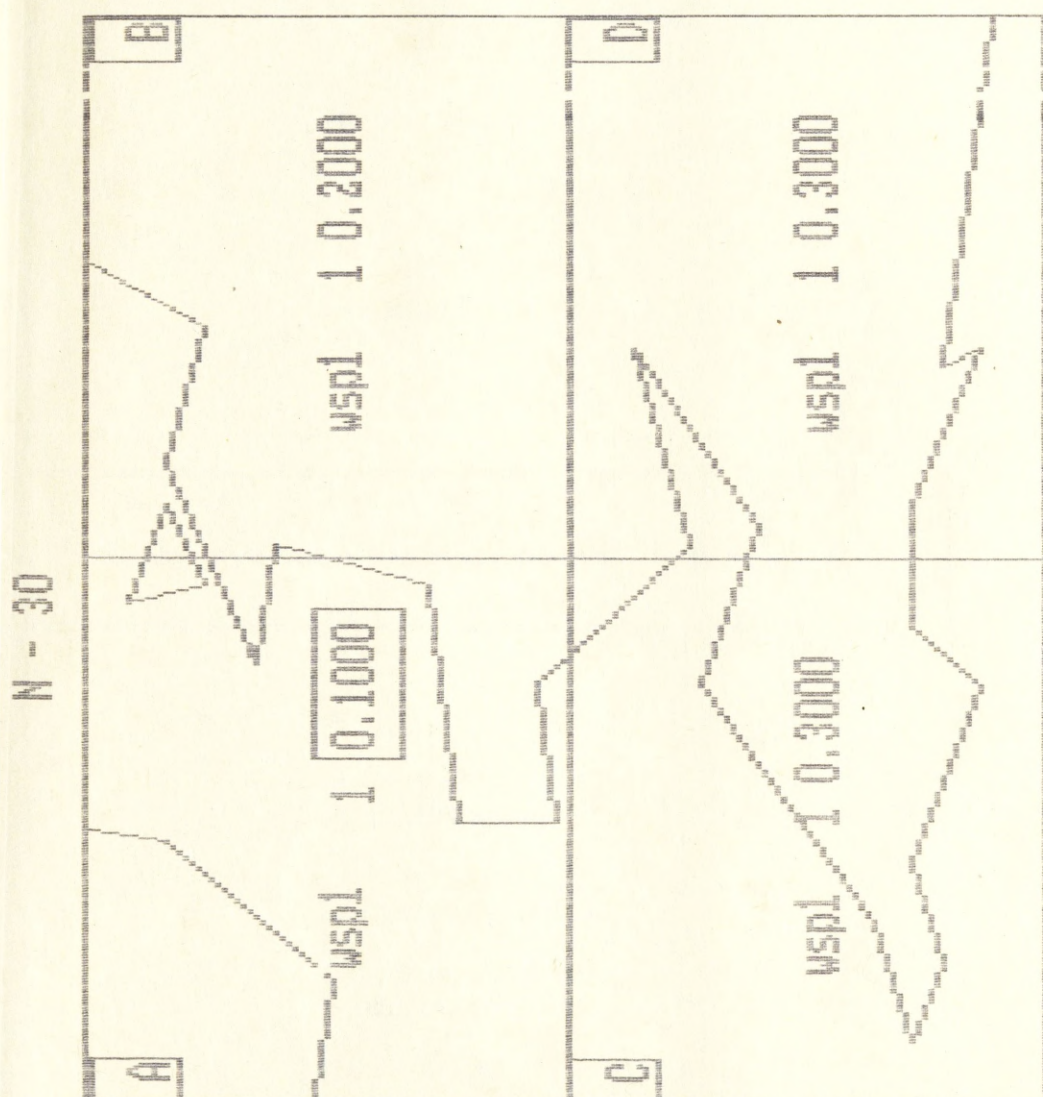
TAK - przechodzimy do wprowadzania dalszych danych.

Po zakończeniu wprowadzenia wszystkich danych dla danej mapy następuje zapisanie ich w zbiorze i przejście do wprowadzania danych związanych z następną mapą.

Zakończenie wprowadzania danych powoduje wprowadzenie danych o ostatniej mapie/1 34/ lub wciśnięcie klawisza F10.

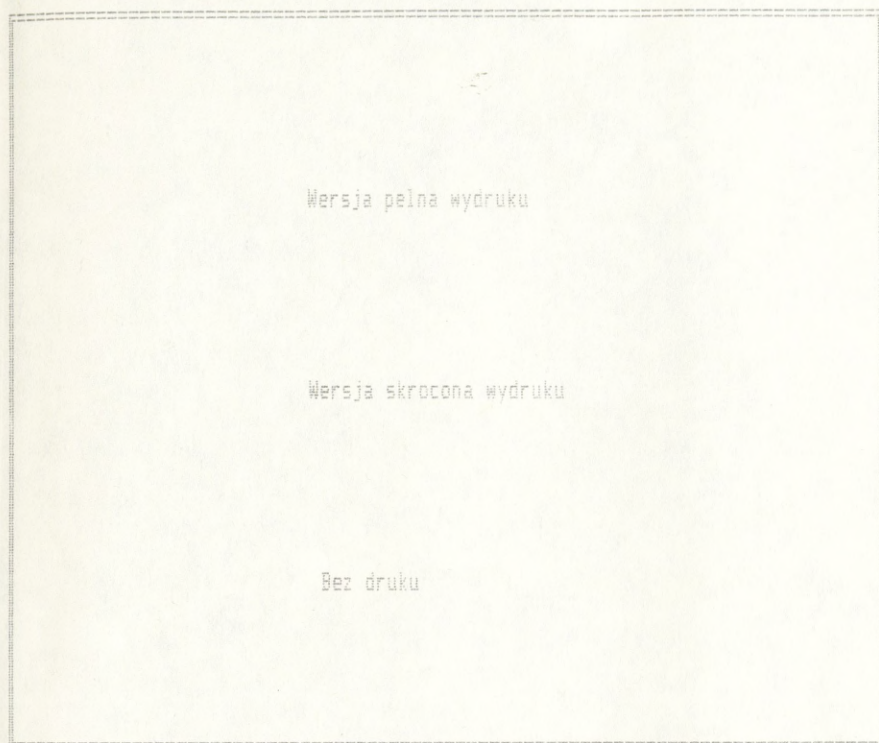
2. Aktualizacja danych

Obsługa tej funkcji jest taka sama jaka funkcji poprzedniej tylko na ekranie wyświetlane są dane ze zbioru dla danej mapy.



3. Wydruk danych

Po wybraniu tej czynności sprawdzana jest sprawność drukarki i dopiero gdy drukarka jest przyłączona, sprawna pojawia się na ekranie

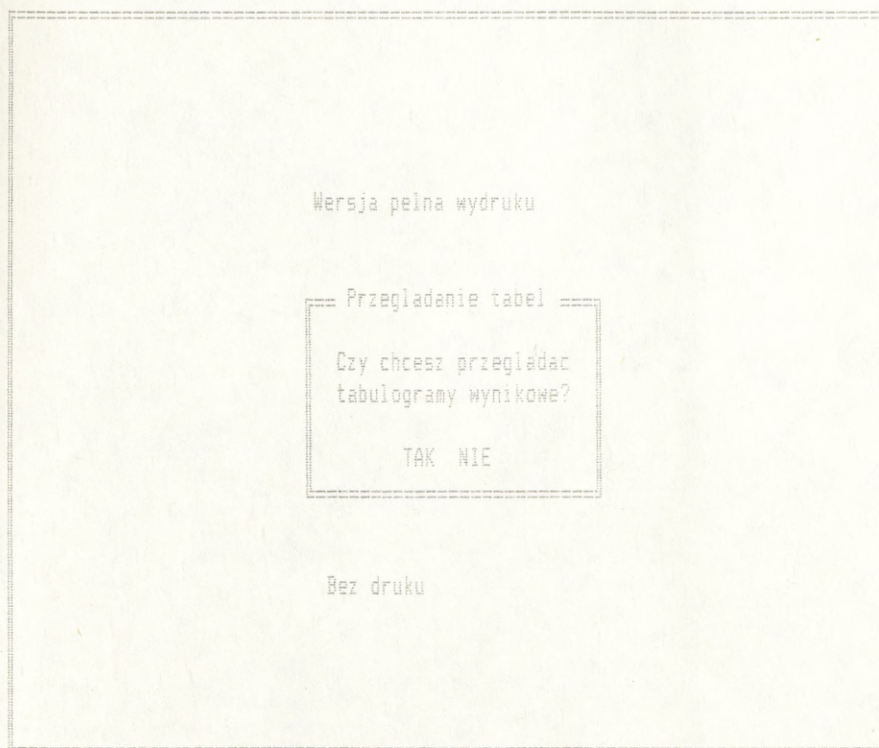


Wersja pełna wydruku

Wersja skrócona wydruku

Bez druku

należy wybrać właściwą postać wydruku



Wersja pełna wydruku

Przeglądanie tabel

Czy chcesz przeglądać tabulogramy wynikowe?

TAK NIE

Bez druku

Wybierz odpowiedź

oraz zdecydować czy chcemy oglądać wyniki na ekranie czy nie.

WSPÓLCZYNNIKI /obliczone wartosci/

Godło mapy	WSPÓLCZYNNIKI																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M_32_B	1	.89	.44	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.88	.4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.9	.64	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.88	.35	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.87	.41	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.89	.52	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.88	.35	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.87	.36	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.86	.29	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
M_33_A	1	.93	.64	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.93	.56	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.94	.48	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.89	.57	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.87	.5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.87	.38	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.85	.23	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.86	.31	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.88	.49	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
M_33_B	1	.93	.41	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.92	.62	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.91	.59	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.87	.31	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.88	.55	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.91	.6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.88	.52	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.85	.32	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.87	.42	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
M_34_A	1	.91	.65	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.91	.69	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.91	.62	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.91	.6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.89	.65	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.9	.52	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.91	.5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.88	.49	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.92	.79	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
M_34_B	1	.89	.77	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.9	.79	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.91	.61	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.86	.46	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.9	.65	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.94	.8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
M_32_C	1	.88	.41	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	2	.88	.33	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	3	.91	.39	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	4	.9	.52	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	5	.89	.41	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	6	.9	.42	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	8	.88	.17	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	9	.85	.21	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

WSPÓLCZYNNIKI /obliczone wartosci/


```

{$R-} {Range checking off}
{$B-} {Boolean short circuiting off}
{$S+} {Stack checking on}
{$I+} {I/O checking on}
{$V-}
{$N-} {No numeric coprocessor}
{$M 65500,65500,655360} {Turbo 3 default stack and heap}

```

```
program wspomaganie;
```

```
Uses Model,Graph,Crt,Dos,Printer,Dodatek,Pamiec,Inne,MapaEuro,Kolbib,Biblinz,Uprocedury,Marbit1;
```

```

const Nws : array [1..20] of string[6]=('wsp1 ', 'wsp2 ', 'wsp3 ', 'wsp4 ',
                                         'wsp5 ', 'wsp6 ', 'wsp7 ', 'wsp8 ', 'wsp9 ', 'wsp10 ',
                                         ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ');
skal: string = ('1 : 1000000 1 : 500000 1 : 200000 1 : 100000 1 : 50000 ');
wydr: string = ('Wersja pelna wydruku  Wersja skrocona wydruku Bez druku ');
czyn: string = ('Wprowadzanie danych Aktualizacja danych Wyprowadzanie danychKoniec dzialania ');
kd : array [1..2] of integer=(1,2);
Xlr : array [1..4] of byte=(30,65,1,1);
Ylr : array [1..4] of byte=(79,67,1,1);
ds : string[3] = 'a:\';
lwsp=20;
dwek=10;
NarGorLewX='0';
NarGorLewY=30;
NarDolPrX='L';
NarDolPrY=35;

```

```
type
```

```

st78 = string[78];
st2 = string[2];
st20 = string[20];
TT = array [1..20,1..10] of integer;
nazpar = function(a,b :byte):byte;
nazpar1 = function(a :char; b :byte):byte;
elem = record
    kw : char;
    wsp : array [1..20] of real;
    T : TT;
end;
XY = array [1..2] of integer;

```

```

Var nak,Nkw,nae1,Ek1,nae2,Ek2,Nkw1,Ek11,Ek21,k0,k1,k2,k3,k4,k5,men,wyd,i,m,n :byte;
GraphMode,GraphDriver,i,j,k,ii,jj,kk : integer;
na,NN,nael,Elkw,NN11,Elkw1,nael1,c1,c11,lan :st2;
f,f1,f2,cz: boolean;
l11,l12 : string[15 ];
wspol : array [1..3,1..3] of XY;
Kratka :array [1..2] of real;
wsob : array [1..80,1..20] of real;
lan1 : string[3];
lanc : string;
wss : array [1..3,1..3,1..20] of real;
wsk : array[1..3,1..3,1..20,1..10] of real;
eel : elem;
inf : ViewPortType;
TB : array [1..3,1..3] of TT;
GM : array [1..80] of st20;
sk : byte;
ch : char;
nn1: nazpar;
nn2: nazpar1;
wyb,wsc: boolean;
zb : file of elem;
qq : real;
pl : text;
lista: FilePtr;

```

```
{$I czolowka.pas}
```

```

procedure wsp(i,j :integer; var l :integer; var f :boolean; var inf : ViewPortType);
var fl : boolean;
    k :st2;
begin l:=1; fl:=false; f:=false;
    while (l<=10) and not fl and not f do begin
        with inf do SetViewPort(x1-2*lit[1],y1,x2,y2,ClipOn); ClearViewPort;
        Str(l,k); OutTextXY(0,1,k); with inf do SetViewPort(x1,y1-2,x2,y2,ClipOn);
        ClearViewPort; ramka(0,0,1); with inf do SetViewPort(x1+2,y1+1,x2,y2,ClipOn);
        wsk[ii,jj,kk,l]:=liczbaR(0.0,1.0,0.0,6,wsk[ii,jj,kk,l],fl,f);
        with inf do SetViewPort(x1,y1-2,x2,y2,ClipOn); ClearViewPort; FullPort;
        if not fl and not f then Inc(l);
        if wsk[ii,jj,kk,l-1]=99.99 then begin
            with inf do SetViewPort(0,0,10*lit[1]+2,10+3*lit[2],ClipOn); OutTextXY(2,2,'wspolczyn. ');
            OutTextXY(4,6+lit[2],'obliczony '); OutTextXY(2,8+2*lit[1],'wczesniej ');
            Rectangle(0,0,10*lit[1]+2,10+3*lit[2]);
            with inf do SetViewPort(x1-2*lit[1],y1+1,x2,y2,ClipOn);
            wss[ii,jj,kk]:=liczbaR(0.0,1.0,0.0,6,wss[ii,jj,kk],fl,f); l:=ii;
            with inf do SetViewPort(0,0,10*lit[1]+2,10+3*lit[2],ClipOn); ClearViewPort; FullPort end;
        end;
    end;
end;

procedure Zapis(cc :char);
var i,j,k,m :byte;
begin
    if SizeD(ds[1])<SizeOf(elem) then begin
        ClrScr; GotoXY(2,13);
        kom25(' BRAK MIEJSCA NA DYSKU - podluz dyskietke i naciśnij dowolny klawisz... ',16,ch,f) end;
        Rewrite(zb);
        for i:=1 to 3-integer(sk<>3) do
            for j:=1 to 3-integer(sk<>3) do begin
                with eel do begin
                    kw:=char(Ord(cc)+j-1+(i-1)*(3-integer(sk<>3)));
                    for k:=1 to 20 do begin wsp[k]:=wss[i,j,k];
                        for m:=1 to 10 do T[k,m]:=Round(wsk[i,j,k,m]*10000) end;
                    write(zb,eel) end end; Close(zb);
            end;
end;

procedure Form(a0,b0,Mx,My,l :word);
var a,b :real;
    i : byte;
begin
    Rectangle(a0,b0,Mx,My); a:=(Mx-a0)/l; b:=(My-b0)/l;
    for i:=1 to l-1 do begin
        line(Round(a0+a*i),b0,Round(a0+a*i),My);
        line(a0,Round(b0+b*i),Mx,Round(b0+b*i)) end;
end;

procedure Ozn(a :byte; a0,b0,Mx,My,l :word; f :boolean);
var b,c,i,j :byte;
    d :word;
    lanc,k2 :string;
begin
    if f then begin b:=3*lit[1]; c:=1 end else begin b:=3; c:=1 end; str(1,k2);
    for j:=1 to l do
        for i:=1 to l do begin
            d:=((Mx-a0) div l)*(i-integer(f and (i mod 2=0))-integer(not f and (i<1)))-
                (lit[2]+8)*integer((not f and (i=1)) or (f and (i mod 2<>0)));
            NazwKw(a0+d,b0+c+((My-b0) div l)*(j-1),8-integer(f)*2,6-integer(f)*5,char(a+i-1+(j-1)*l));
            Str(wsk[j,i,1,1]:2:4,lanc); wspol[j,i,1]:=a0+((Mx-a0) div l)*(i-1)+b;
            wspol[j,i,2]:=b0+((My-b0) div l)*(j-1)+(My-c-i*lit[2]) div (2*1); FullPort;
            OutTextXY(Wspol[j,i,1],Wspol[j,i,2],Nws[i]);
            OutTextXY(Wspol[j,i,1]+Length(Nws[i])*lit[1]+4+4*integer(not f),Wspol[j,i,2],k2);
            OutTextXY(Wspol[j,i,1]+(Length(Nws[i])+Length(k2))*lit[1]+8+8*integer(not f),Wspol[j,i,2],lanc) end;
        end;
end;

function ciag(sk :byte):string;
begin
    case sk of
        1,4 : ciag:='A';
    end;
end;

```

```

5   : ciag:='a';
end end;

procedure Czyt(m : byte; var i :byte);
var k1 :byte;
begin
  repeat
    Read(zb,eel); GM[i]:=ConCat(copy(l11,1,length(l11)-4), ' ',eel,kw);
    if eel.kw>='A' then ii:=Ord(UpCase(eel.kw))+1-Ord('A');
    with eel do begin
      for kk:=1 to lwsp do begin
        case m of
          1 : wss[wier(ii,3-integer(sk<>3))+1,Kol(ii,3-integer(sk<>3))+1,kk]:=wsp[kk];
          2 : wsob[i,kk]:=wsp[kk];
        end;
        for k1:=1 to dwek do
          wsk[wier(ii,3-integer(sk<>3))+1,Kol(ii,3-integer(sk<>3))+1,kk,k1]:=T[kk,k1]/10000 end end;
        Inc(i);
      until EOF(zb); Dec(i)
    end;
end;

function Lgodlo(i,j,k,l,m,n :byte; f :boolean):string;
var lan1,lan2 :string[30];
    kre : string[3];
begin
  if f then kre:=' - ' else kre:=' _ ';
  str(j,lan1); lan2:=ConCat(char(i)+kre+lan1); lan1:= ' ';
  if (sk>1) or ((sk=0) and (i<>0)) then lan2:=ConCat(lan2,kre,chr(k)); lan2:=ConCat(lan2,lan1); lan1:= ' ';
  if (sk>2) or ((sk=0) and (m<>0)) then str(l,lan1); lan2:=ConCat(lan2,lan1); lan1:= ' ';
  if (sk>3) or ((sk=0) and (n<>32) and (n<>0)) then str(m,lan1); lan2:=ConCat(lan2,lan1);
  if (sk>4) then lan2:=ConCat(lan2,' ',chr(n)); Lgodlo:=lan2
end;

procedure Wprow(i,j,k,l,m,n : integer);
var nn : byte;
begin
  {#F+}
  xpcz:=Round((Nar(j,k,l,m,Kol)-Nar(X1r[1],X1r[2],X1r[3],X1r[4],Kol))*Kratka[1]);
  ypcz:=Round((Nar(Y1r[1]-1,Ord(k),l,m,Wier)-Nar(0,Y1r[2],Y1r[3],Y1r[4],Wier))*Kratka[2]);
  {#F-}
  skalay:=(MaxY-4*lit[2])/Kratka[2];    skalax:=((MaxY-4*lit[2])/q)/Kratka[1];
  lx1:=Round((MaxX-Kratka[1]*skalax)/2); ly1:=2*lit[2];
  lx2:=MaxX-lx1;    ly2:=MaxY-4*lit[2];
  mapanr(10); delay(800); l11:=Lgodlo(i,j,k,l,m,n,false);
  l12:=l11+'.N10'; l11:=l11+'.NUM';
  OutTextXY(0,MaxY-10,' Enter - koniec pola END - koniec wektora sklad. ESC - koniec wspolczynnikow');
  for ii:=1 to 4 do for jj:=1 to 4 do for kk:=1 to lwsp do begin
    for nn:=1 to dwek do wsk[ii,jj,kk,nn]:=0.0; wss[ii,jj,kk]:=0.0 end;
  Assign(zb,l11);
  if FileExist(l11) then begin Reset(zb);
  if FileSize(zb)<>0 then begin Seek(zb,0);
  Czyt(1,nn) end end
  else begin for ii:=1 to 3 do for jj:=1 to 3 do
    for kk:=1 to lwsp do for k1:=1 to dwek do
      with eel do TB[ii,jj,kk,k1]:=0; Rewrite(zb) end;
    Form(lx1,ly1,lx2,ly2,2+integer(sk=2)); lan1:=ciag(sk);
    Gz(ord(lan1[1]),lx1,ly1,lx2,ly2,2+integer(sk=2),sk<>2);
    FullPort; f:=false; lan2:=Lgodlo(i,j,k,l,m,n,true);
    OutTextXY((MaxX-13*lit[1]) div 2,lit[2] div 2,lan2); i:=(length(Nws[1])+2)*lit[1];
    for ii:=1 to 2+integer(sk=2) do for jj:=1 to 2+integer(sk=2) do begin kk:=i; f2:=false;
    while (kk<lwsp) and not f2 do begin
      if not f then begin ch:='N';
      while ch='N' do begin
        inf.x1:=Wspol[ii,jj,1]+i;    inf.y1:=Wspol[ii,jj,2]-1;
        inf.x2:=inf.x1+6*lit[2]+8;    inf.y2:=inf.y1+lit[2]+6; l:=1; f1:=false;
        SetViewPort(Wspol[ii,jj,1],Wspol[ii,jj,2],Wspol[ii,jj,1]+15*lit[1],inf.y2,ClipOn);
        ClearViewPort; OutTextXY(2,0,Nws[kk]); wsp(ii,jj,l,f2,inf); FullPort;
        with inf do SetViewPort(x1-(2*lit[1])*integer(wsk[ii,jj,kk,1]>99.99)),y1,x2,y2,ClipOn); ClearViewPort;
        if wsk[ii,jj,kk,1]>99.99 then begin
          wss[ii,jj,kk]:=0.0; for m:=1 to l do wss[ii,jj,kk]:=wss[ii,jj,kk]+wsk[ii,jj,kk,m];
          wss[ii,jj,kk]:=wss[ii,jj,kk]/l end else wsk[ii,jj,kk,1]:=0.0;
          Str(wss[ii,jj,kk]:4:4,l12); OutTextXY(2,l12); FullPort;
        end;
      end;
    end;
  end;
end;

```

```

SetViewport(0,MaxY-10,MaxX,MaxY,ClipOn); ClearViewport; FullPort;
Komunikat6('Czy prawdziwa wartosc?',3,q,ch);
SetViewport(0,MaxY-10,MaxX,MaxY,ClipOn); ClearViewport; FullPort;
OutTextXY(0,MaxY-10,' Enter - koniec pola END - koniec wektora sklad. ESC - koniec wspolczynnikow');
with inf do SetViewport(x1,y1,x2,y2,ClipOn); ramka(0,0,0); end; Inc(kk); FullPort;
if not f2 then begin
  SetViewport(Wspol[ii,jj,1],Wspol[ii,jj,2],Wspol[ii,jj,1]+15*lit[1],inf.y2,ClipOn); ClearViewport;
  Str(wsk[ii,jj,1]:1:4,lanc); FullPort;
  OutTextXY(Wspol[ii,jj,1],Wspol[ii,jj,2],Nws[kk]);
  OutTextXY(Wspol[ii,jj,1]+Length(Nws[1])*lit[1]+4*integer(not f),Wspol[ii,jj,2],'1');
  OutTextXY(Wspol[ii,jj,1]+(Length(Nws[1])+2)*lit[1]+8*integer(not f),Wspol[ii,jj,2],lanc) end;
end end end end;

```

```

procedure dbl(var blad:boolean);
begin
  dzwiek; blad:=true;
end;

```

```

procedure WydrukP;
const nagl:array[1..7] of string=(

```

Godlo	mapy	wspolczyn.	SKLADNIKI											
			1	2	3	4	5	6	7	8	9	10		

```

tytul:string='wspolczyniki';
str0:array[1..5,1..2] of byte=((21,71),(32,87),(40,95),(48,103),(56,111));
obk:opklaw=('0'..'PgUp','PgDn','N','F','Home','End','Esc',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ');
obt:opklaw=('WIERSZE','KOLUMNY','KONIEC PRZEGLADANIA TABELI',' ',' ',' ',' ',' ',' ',' ');
niwspol : byte=0;

```

```

var zn:char;
enter,blad,func:boolean;
str,ofwrsz:byte;

```

```

procedure pizszi(i,j,w,ye,ww,pocz,kon,druk:byte);

```

```

var lin :string;
k :byte;

```

```

begin
  lin:=ziag(sk); k:=Ord(lin[1]); lin:=nagl[6];
  if sk<>0 then if (w=1) or (ye=9) then GM[i]:=ConCat(copy(lin,1,length(lin)-4),' ',chr(k+(i-1)*(2+integer(sk=3))+j-1))
  else GM[i]:='';
  if (w=1) or (ye=9) then pizsstring(GM[i+(ww div lwsz)*integer(sk=0)],lin,2,18);
  pizsbyte(w,lin,18,2); pizsreal(wes[i,j,w],8,6,lin,21,10);
  for k:=1 to dwak do pizsreal(wsk[i,j,w,k],6,4,lin,32+(k-1)*8,7);
  pizsdwlin(20,pocz,kon,ye,druk,lin,1st); lin:=nagl[6];
end;

```

```

procedure tresk(pocz,konw,pocz,konk,druk:byte);

```

```

var i:byte;
begin
  for i:=1 to 5 do pizsdwlin(20,pocz,konk,i+3,druk,nagl[i],1st);
  for i:=pocz to konw do
    pizszi(wier(((i-integer(i mod lwsz=0)) div lwsz)+1,2+integer(sk=3))+1,
    Kol(((i-integer(i mod lwsz=0)) div lwsz)+1,2+integer(sk=3))+1,
    (i mod lwsz)+integer(i mod lwsz=0)*lwsz,i-pocz+9,i,pocz,konk,druk);
  if druk=1 then pizsdwlin(20,pocz,konk,89,druk,nagl[7],1st);
end;

```

```

begin

```

```

  if oio then begin
    kole; bezobw(2,3,79,23); opisklaw(obk,obt);delay(1000);
    liniegorne(3,' WSPOLCZYNNIKI ');
    str:=i; ofwrsz:=0; blad:=false; enter:=false;
    repeat
      n:=(ofwrsz+16)*integer(ofwrsz<65)+80*integer(ofwrsz>=65);
      if not blad then begin
        tresk(nofwrsz+16,nofwrsz+16,nofwrsz+16,nofwrsz+16);

```

```

    strzalki(1,0,80,15,str,5,21,str0[1]-str0[2]+22,7);
end;
blad:=false; czytczyn(zn,func);
if func then
    case zn of
        #75:if str<>1 then str:=str-1 else dbl(blad);
        #77:if str<>5 then str:=str+1 else dbl(blad);
        #71:if str<>1 then str:=1 else dbl(blad);
        #79:if str<>5 then str:=5 else dbl(blad);
        #72:if ofwrsz<>0 then ofwrsz:=ofwrsz-1 else dbl(blad);
        #80:if ofwrsz<65 then ofwrsz:=ofwrsz+1 else dbl(blad);
        #73:if ofwrsz>16 then ofwrsz:=(ofwrsz-16)*integer(ofwrsz>16) else begin ofwrsz:=0; dbl(blad); blad:=false end;
        #81:if ofwrsz<65 then ofwrsz:=ofwrsz+16 else begin ofwrsz:=65; dbl(blad); blad:=false end;
        else dbl(blad) end
    else if zn=#27 then enter:=true else dbl(blad)
until enter;
end;
if nlwspol<>0 then for i:=1 to 0 do writeln(1st,tytul[i]);
writeln(1st,tytul); tresc(1,80,21,111,1);
end;

```

```

procedure WydrukS(n :byte);
const nagl:array[1..7] of string=(

```

WSPOLCZYNNIKI											
Godło mapy	1	2	3	4	5	6	7	8	9	10	11
			12	13	14	15	16	17	18	19	20

```

tytul:string='
                                W S P O L C Z Y N N I K I /obliczone wartosci/';
str0:array [1..14,1..2] of byte=((21,76),(29,84),(37,92),(45,100),(53,108),(61,116),(69,124),(77,132),
                                (85,140),(93,148),(101,156),(109,164),(117,172),(125,180));
obk:opklaw=('0' PgUp PgDn ',' N T Home End ',' Esc ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ');
obt:opklaw=(' WIERSZE ',' KOLUMNY ',' KONIEC PRZEGLADANIA TABELI ',' ',' ',' ',' ',' ',' ',' ',' ');

```

```

var zn          :char;
    enter,blad,func:boolean;
    str,ofwrsz,nl :byte;

```

```

procedure pizsln(i,ye,pocz,kon,druk:byte);
var lin,gdm:string;
    k,l :byte;
begin
    lin:=nagl[6]; l:=0;
    for k:=1 to Length(GM[i]) do
        if GM[i,k]=gdm[k] then Inc(l);
    if not (l=length(gdm)-1) or (i=1) then gdm:=GM[i] else gdm:=Concat(Copy(lin,2,1),GM[i,1+1]);
    pizstring(gdm,lin,2,18); gdm:=GM[i];
    for k:=1 to lwsp do
        pizsreal(wsob[i,k],6,4,lin,21+(k-1)*8,7);
    pizsdowlin(20,pocz,kon,ye,druk,lin,1st);
end;

```

```

procedure tresc(pocz,konw,pocz, Konk,druk:byte);
var i:byte;
begin
    for i:=1 to 5 do pizsdowlin(20,pocz, Konk,i+3,druk,nagl[i],1st);
    for i:=pocz to konw do pizsln(i,i-pocz+9,pocz, Konk,druk);
    if druk=1 then pizsdowlin(20,pocz, Konk,89,druk,nagl[7],1st);
end;

```

```

begin
    if olo then begin kole; bezobw(2,3,79,23); opisklaw(obk,obt);
    lin:=nagl[3]; wsob:=nagl[4]; obliczone_wartosci;

```

```

str:=1; ofwrsz:=0; blad:=false; enter:=false; nn:=15*integer(n>15)+n*integer(n<=15);
repeat
  if not blad then begin
    tresc(ofwrsz+1,ofwrsz+nn,str0[str,1],str0[str,2],0);
    strzalki(1,0,80,15,str,14,21,str0[str,2]-str0[str,1]+22,9);
  end;
  blad:=false; czytczyn(zn,func);
  if func then
    case zn of
      #75:if str<>1 then str:=str-1 else dbl(blad);
      #77:if str<>14 then str:=str+1 else dbl(blad);
      #71:if str<>1 then str:=1 else dbl(blad);
      #79:if str<>14 then str:=14 else dbl(blad);
      #72:if ofwrsz<>0 then ofwrsz:=ofwrsz-1 else dbl(blad);
      #80:if ofwrsz<n-15 then ofwrsz:=ofwrsz+1 else dbl(blad);
      #73:if ofwrsz<>0 then ofwrsz:=ofwrsz-15*integer(ofwrsz>15) else begin ofwrsz:=0; dbl(blad); blad:=false end;
      #81:if ofwrsz<n-15 then ofwrsz:=ofwrsz+15*integer(ofwrsz<ofwrsz-15)
        else begin ofwrsz:=n-15; dbl(blad); blad:=false end;
    else dbl(blad)
    end
  else
    if zn=#27 then enter:=true else dbl(blad)
  until enter;
end;
{
  if niwydrsk<>0 then for i:=1 to 0 do writeln(1st,tytul[i]);
  nn:=80*integer(n>=80)+n*integer(n<80);
  writeln(1st,tytul); tresc(1,nn,21,180,1);
end;

function koniec(na : st2; nak :byte; nael : st2; nael,nae2 : byte; naci :st2;
  NN11 : st2; Nkw1 : byte; Eikw1 : st2; Ek11,Ek21 : byte; cli : st2; as :nazpar; d :boolean):boolean;
begin
  if (na=NN11) or d then
    if (nak=Nkw1) or not d then
      if nael='' then koniec:=true else begin
        if as(Ord(nael[1]),2)=as(Ord(Eikw1[1]),2) then
          if nael=0 then koniec:=true else begin
            if as(nael,3)=as(Ek11,3) then
              if nae2=0 then koniec:=true else begin
                if as(nae2,2)=as(Ek21,2) then
                  if cli='' then koniec:=true else begin
                    if as(Ord(cli[1]),2)=as(Ord(naci[1]),2) then koniec:=true
                  else koniec:=false end
                else koniec:=false end
              else koniec:=false end
            else koniec:=false end
          else koniec:=false
        else koniec:=false
      end;

procedure krok(var N :st2; var K1 :byte; var E1 :st2; var K9,K4 :byte; var c1 :st2);
begin
  if (c1<>'') and ((Ord(c1[1])-64) mod 2=1) then c1[1]:=chr(Ord(c1[1])+1) else begin if c1<>' ' then c1[1]:=chr(Ord(c1[1])-1);
  if (K4<>0) and (K4 mod 2=1) then K4:=K4+1 else begin if K4<>0 then K4:=K4-1;
  if (K9<>0) and (K9 mod 3<2) and (K9 mod 3<>0) then K9:=K9+1 else begin if K9<>0 then K9:=K9-2;
  if (E1<>'') and ((Ord(E1[1])-64) mod 2=1) then E1[1]:=chr(Ord(E1[1])+1) else begin
  if E1<>' ' then E1[1]:=chr(Ord(E1[1])-1);
  K1:=K1+1 end
end end end end;

begin
  Inicjacja(GraphDriver,GraphMode); skaikx:=MaxX/639; Skalky:=MaxY/199; RestorecrtMode;
  TextBackground(green); Textcolor(0); Cursor(OFF); Czolowka; ChDir(ds);
  while true do begin
    men:=Menu(4,czyn,''); wyb:=true;
    while wyb do begin
      case men of
        1,2 : begin
          sk:=Menu(5,skal,' Wybiez skale mapy '); xpocz:=0; ypocz:=0; Skalax:=1.0; Skalay:=1.0;
          SetGraphMode(GetGraphMode); mapanr(0); siatka(sk); lani:=ReadKey;
          SetGraphMode(GetGraphMode); FullPort; ClearDevice;
          Gdln(NN #kx Ekx Ek1 Ek2 #f #0 #1 #2 #3 #4 #5 #6 #7 #8 #9 #10 #11 #12 #13 #14 #15 #16 #17 #18 #19 #20 #21 #22 #23 #24 #25 #26 #27 #28 #29 #30 #31 #32 #33 #34 #35 #36 #37 #38 #39 #40 #41 #42 #43 #44 #45 #46 #47 #48 #49 #50 #51 #52 #53 #54 #55 #56 #57 #58 #59 #60 #61 #62 #63 #64 #65 #66 #67 #68 #69 #70 #71 #72 #73 #74 #75 #76 #77 #78 #79 #80 #81 #82 #83 #84 #85 #86 #87 #88 #89 #90 #91 #92 #93 #94 #95 #96 #97 #98 #99 #100 #101 #102 #103 #104 #105 #106 #107 #108 #109 #110 #111 #112 #113 #114 #115 #116 #117 #118 #119 #120 #121 #122 #123 #124 #125 #126 #127 #128 #129 #130 #131 #132 #133 #134 #135 #136 #137 #138 #139 #140 #141 #142 #143 #144 #145 #146 #147 #148 #149 #150 #151 #152 #153 #154 #155 #156 #157 #158 #159 #160 #161 #162 #163 #164 #165 #166 #167 #168 #169 #170 #171 #172 #173 #174 #175 #176 #177 #178 #179 #180 #181 #182 #183 #184 #185 #186 #187 #188 #189 #190 #191 #192 #193 #194 #195 #196 #197 #198 #199 #200 #201 #202 #203 #204 #205 #206 #207 #208 #209 #210 #211 #212 #213 #214 #215 #216 #217 #218 #219 #220 #221 #222 #223 #224 #225 #226 #227 #228 #229 #230 #231 #232 #233 #234 #235 #236 #237 #238 #239 #240 #241 #242 #243 #244 #245 #246 #247 #248 #249 #250 #251 #252 #253 #254 #255 #256 #257 #258 #259 #260 #261 #262 #263 #264 #265 #266 #267 #268 #269 #270 #271 #272 #273 #274 #275 #276 #277 #278 #279 #280 #281 #282 #283 #284 #285 #286 #287 #288 #289 #290 #291 #292 #293 #294 #295 #296 #297 #298 #299 #300 #301 #302 #303 #304 #305 #306 #307 #308 #309 #310 #311 #312 #313 #314 #315 #316 #317 #318 #319 #320 #321 #322 #323 #324 #325 #326 #327 #328 #329 #330 #331 #332 #333 #334 #335 #336 #337 #338 #339 #340 #341 #342 #343 #344 #345 #346 #347 #348 #349 #350 #351 #352 #353 #354 #355 #356 #357 #358 #359 #360 #361 #362 #363 #364 #365 #366 #367 #368 #369 #370 #371 #372 #373 #374 #375 #376 #377 #378 #379 #380 #381 #382 #383 #384 #385 #386 #387 #388 #389 #390 #391 #392 #393 #394 #395 #396 #397 #398 #399 #400 #401 #402 #403 #404 #405 #406 #407 #408 #409 #410 #411 #412 #413 #414 #415 #416 #417 #418 #419 #420 #421 #422 #423 #424 #425 #426 #427 #428 #429 #430 #431 #432 #433 #434 #435 #436 #437 #438 #439 #440 #441 #442 #443 #444 #445 #446 #447 #448 #449 #450 #451 #452 #453 #454 #455 #456 #457 #458 #459 #460 #461 #462 #463 #464 #465 #466 #467 #468 #469 #470 #471 #472 #473 #474 #475 #476 #477 #478 #479 #480 #481 #482 #483 #484 #485 #486 #487 #488 #489 #490 #491 #492 #493 #494 #495 #496 #497 #498 #499 #500 #501 #502 #503 #504 #505 #506 #507 #508 #509 #510 #511 #512 #513 #514 #515 #516 #517 #518 #519 #520 #521 #522 #523 #524 #525 #526 #527 #528 #529 #530 #531 #532 #533 #534 #535 #536 #537 #538 #539 #540 #541 #542 #543 #544 #545 #546 #547 #548 #549 #550 #551 #552 #553 #554 #555 #556 #557 #558 #559 #560 #561 #562 #563 #564 #565 #566 #567 #568 #569 #570 #571 #572 #573 #574 #575 #576 #577 #578 #579 #580 #581 #582 #583 #584 #585 #586 #587 #588 #589 #590 #591 #592 #593 #594 #595 #596 #597 #598 #599 #600 #601 #602 #603 #604 #605 #606 #607 #608 #609 #610 #611 #612 #613 #614 #615 #616 #617 #618 #619 #620 #621 #622 #623 #624 #625 #626 #627 #628 #629 #630 #631 #632 #633 #634 #635 #636 #637 #638 #639 #640 #641 #642 #643 #644 #645 #646 #647 #648 #649 #650 #651 #652 #653 #654 #655 #656 #657 #658 #659 #660 #661 #662 #663 #664 #665 #666 #667 #668 #669 #670 #671 #672 #673 #674 #675 #676 #677 #678 #679 #680 #681 #682 #683 #684 #685 #686 #687 #688 #689 #690 #691 #692 #693 #694 #695 #696 #697 #698 #699 #700 #701 #702 #703 #704 #705 #706 #707 #708 #709 #710 #711 #712 #713 #714 #715 #716 #717 #718 #719 #720 #721 #722 #723 #724 #725 #726 #727 #728 #729 #730 #731 #732 #733 #734 #735 #736 #737 #738 #739 #740 #741 #742 #743 #744 #745 #746 #747 #748 #749 #750 #751 #752 #753 #754 #755 #756 #757 #758 #759 #760 #761 #762 #763 #764 #765 #766 #767 #768 #769 #770 #771 #772 #773 #774 #775 #776 #777 #778 #779 #780 #781 #782 #783 #784 #785 #786 #787 #788 #789 #790 #791 #792 #793 #794 #795 #796 #797 #798 #799 #800 #801 #802 #803 #804 #805 #806 #807 #808 #809 #810 #811 #812 #813 #814 #815 #816 #817 #818 #819 #820 #821 #822 #823 #824 #825 #826 #827 #828 #829 #830 #831 #832 #833 #834 #835 #836 #837 #838 #839 #840 #841 #842 #843 #844 #845 #846 #847 #848 #849 #850 #851 #852 #853 #854 #855 #856 #857 #858 #859 #860 #861 #862 #863 #864 #865 #866 #867 #868 #869 #870 #871 #872 #873 #874 #875 #876 #877 #878 #879 #880 #881 #882 #883 #884 #885 #886 #887 #888 #889 #890 #891 #892 #893 #894 #895 #896 #897 #898 #899 #900 #901 #902 #903 #904 #905 #906 #907 #908 #909 #910 #911 #912 #913 #914 #915 #916 #917 #918 #919 #920 #921 #922 #923 #924 #925 #926 #927 #928 #929 #930 #931 #932 #933 #934 #935 #936 #937 #938 #939 #940 #941 #942 #943 #944 #945 #946 #947 #948 #949 #950 #951 #952 #953 #954 #955 #956 #957 #958 #959 #960 #961 #962 #963 #964 #965 #966 #967 #968 #969 #970 #971 #972 #973 #974 #975 #976 #977 #978 #979 #980 #981 #982 #983 #984 #985 #986 #987 #988 #989 #990 #991 #992 #993 #994 #995 #996 #997 #998 #999 #1000 #1001 #1002 #1003 #1004 #1005 #1006 #1007 #1008 #1009 #1010 #1011 #1012 #1013 #1014 #1015 #1016 #1017 #1018 #1019 #1020 #1021 #1022 #1023 #1024 #1025 #1026 #1027 #1028 #1029 #1030 #1031 #1032 #1033 #1034 #1035 #1036 #1037 #1038 #1039 #1040 #1041 #1042 #1043 #1044 #1045 #1046 #1047 #1048 #1049 #1050 #1051 #1052 #1053 #1054 #1055 #1056 #1057 #1058 #1059 #1060 #1061 #1062 #1063 #1064 #1065 #1066 #1067 #1068 #1069 #1070 #1071 #1072 #1073 #1074 #1075 #1076 #1077 #1078 #1079 #1080 #1081 #1082 #1083 #1084 #1085 #1086 #1087 #1088 #1089 #1090 #1091 #1092 #1093 #1094 #1095 #1096 #1097 #1098 #1099 #1100 #1101 #1102 #1103 #1104 #1105 #1106 #1107 #1108 #1109 #1110 #1111 #1112 #1113 #1114 #1115 #1116 #1117 #1118 #1119 #1120 #1121 #1122 #1123 #1124 #1125 #1126 #1127 #1128 #1129 #1130 #1131 #1132 #1133 #1134 #1135 #1136 #1137 #1138 #1139 #1140 #1141 #1142 #1143 #1144 #1145 #1146 #1147 #1148 #1149 #1150 #1151 #1152 #1153 #1154 #1155 #1156 #1157 #1158 #1159 #1160 #1161 #1162 #1163 #1164 #1165 #1166 #1167 #1168 #1169 #1170 #1171 #1172 #1173 #1174 #1175 #1176 #1177 #1178 #1179 #1180 #1181 #1182 #1183 #1184 #1185 #1186 #1187 #1188 #1189 #1190 #1191 #1192 #1193 #1194 #1195 #1196 #1197 #1198 #1199 #1200 #1201 #1202 #1203 #1204 #1205 #1206 #1207 #1208 #1209 #1210 #1211 #1212 #1213 #1214 #1215 #1216 #1217 #1218 #1219 #1220 #1221 #1222 #1223 #1224 #1225 #1226 #1227 #1228 #1229 #1230 #1231 #1232 #1233 #1234 #1235 #1236 #1237 #1238 #1239 #1240 #1241 #1242 #1243 #1244 #1245 #1246 #1247 #1248 #1249 #1250 #1251 #1252 #1253 #1254 #1255 #1256 #1257 #1258 #1259 #1260 #1261 #1262 #1263 #1264 #1265 #1266 #1267 #1268 #1269 #1270 #1271 #1272 #1273 #1274 #1275 #1276 #1277 #1278 #1279 #1280 #1281 #1282 #1283 #1284 #1285 #1286 #1287 #1288 #1289 #1290 #1291 #1292 #1293 #1294 #1295 #1296 #1297 #1298 #1299 #1300 #1301 #1302 #1303 #1304 #1305 #1306 #1307 #1308 #1309 #1310 #1311 #1312 #1313 #1314 #1315 #1316 #1317 #1318 #1319 #1320 #1321 #1322 #1323 #1324 #1325 #1326 #1327 #1328 #1329 #1330 #1331 #1332 #1333 #1334 #1335 #1336 #1337 #1338 #1339 #1340 #1341 #1342 #1343 #1344 #1345 #1346 #1347 #1348 #1349 #1350 #1351 #1352 #1353 #1354 #1355 #1356 #1357 #1358 #1359 #1360 #1361 #1362 #1363 #1364 #1365 #1366 #1367 #1368 #1369 #1370 #1371 #1372 #1373 #1374 #1375 #1376 #1377 #1378 #1379 #1380 #1381 #1382 #1383 #1384 #1385 #1386 #1387 #1388 #1389 #1390 #1391 #1392 #1393 #1394 #1395 #1396 #1397 #1398 #1399 #1400 #1401 #1402 #1403 #1404 #1405 #1406 #1407 #1408 #1409 #1410 #1411 #1412 #1413 #1414 #1415 #1416 #1417 #1418 #1419 #1420 #1421 #1422 #1423 #1424 #1425 #1426 #1427 #1428 #1429 #1430 #1431 #1432 #1433 #1434 #1435 #1436 #1437 #1438 #1439 #1440 #1441 #1442 #1443 #1444 #1445 #1446 #1447 #1448 #1449 #1450 #1451 #1452 #1453 #1454 #1455 #1456 #1457 #1458 #1459 #1460 #1461 #1462 #1463 #1464 #1465 #1466 #1467 #1468 #1469 #1470 #1471 #1472 #1473 #1474 #1475 #1476 #1477 #1478 #1479 #1480 #1481 #1482 #1483 #1484 #1485 #1486 #1487 #1488 #1489 #1490 #1491 #1492 #1493 #1494 #1495 #1496 #1497 #1498 #1499 #1500 #1501 #1502 #1503 #1504 #1505 #1506 #1507 #1508 #1509 #1510 #1511 #1512 #1513 #1514 #1515 #1516 #1517 #1518 #1519 #1520 #1521 #1522 #1523 #1524 #1525 #1526 #1527 #1528 #1529 #1530 #1531 #1532 #1533 #1534 #1535 #1536 #1537 #1538 #1539 #1540 #1541 #1542 #1543 #1544 #1545 #1546 #1547 #1548 #1549 #1550 #1551 #1552 #1553 #1554 #1555 #1556 #1557 #1558 #1559 #1560 #1561 #1562 #1563 #1564 #1565 #1566 #1567 #1568 #1569 #1570 #1571 #1572 #1573 #1574 #1575 #1576 #1577 #1578 #1579 #1580 #1581 #1582 #1583 #1584 #1585 #1586 #1587 #1588 #1589 #1590 #1591 #1592 #1593 #1594 #1595 #1596 #1597 #1598 #1599 #1600 #1601 #1602 #1603 #1604 #1605 #1606 #1607 #1608 #1609 #1610 #1611 #1612 #1613 #1614 #1615 #1616 #1617 #1618 #1619 #1620 #1621 #1622 #1623 #1624 #1625 #1626 #1627 #1628 #1629 #1630 #1631 #1632 #1633 #1634 #1635 #1636 #1637 #1638 #1639 #1640 #1641 #1642 #1643 #1644 #1645 #1646 #1647 #1648 #1649 #1650 #1651 #1652 #1653 #1654 #1655 #1656 #1657 #1658 #1659 #1660 #1661 #1662 #1663 #1664 #1665 #1666 #1667 #1668 #1669 #1670 #1671 #1672 #1673 #1674 #1675 #1676 #1677 #1678 #1679 #1680 #1681 #1682 #1683 #1684 #1685 #1686 #1687 #1688 #1689 #1690 #1691 #1692 #1693 #1694 #1695 #1696 #1697 #1698 #1699 #1700 #1701 #1702 #1703 #1704 #1705 #1706 #1707 #1708 #1709 #1710 #1711 #1712 #1713 #1714 #1715 #1716 #1717 #1718 #1719 #1720 #1721 #1722 #1723 #1724 #1725 #1726 #1727 #1728 #1729 #1730 #1731 #1732 #1733 #1734 #1735 #1736 #1737 #1738 #1739 #1740 #1741 #1742 #1743 #1744 #1745 #1746 #1747 #1748 #1749 #1750 #1751 #1752 #1753 #1754 #1755 #1756 #1757 #1758 #1759 #1760 #1761 #1762 #1763 #1764 #1765 #1766 #1767 #1768 #1769 #1770 #1771 #1772 #1773 #1774 #1775 #1776 #1777 #1778 #1779 #1780 #1781 #1782 #1783 #1784 #1785 #1786 #1787 #1788 #1789 #1790 #1791 #1792 #1793 #1794 #1795 #1796 #1797 #1798 #1799 #1800 #1801 #1802 #1803 #1804 #1805 #1806 #1807 #1808 #1809 #1810 #1811 #1812 #1813 #1814 #1815 #1816 #1817 #1818 #1819 #1820 #1821 #1822 #1823 #1824 #1825 #1826 #1827 #1828 #1829 #1830 #1831 #1832 #1833 #1834 #1835 #1836 #1837 #1838 #1839 #1840 #1841 #1842 #1843 #1844 #1845 #1846 #1847 #1848 #1849 #1850 #1851 #1852 #1853 #1854 #1855 #1856 #1857 #1858 #1859 #1860 #1861 #1862 #1863 #1864 #1865 #1866 #1867 #1868 #1869 #1870 #1871 #1872 #1873 #1874 #1875 #1876 #1877 #1878 #1879 #1880 #1881 #1882 #1883 #1884 #1885 #1886 #1887 #1888 #1889 #1890 #1891 #1892 #1893 #1894 #1895 #1896 #1897 #1898 #1899 #1900 #1901 #1902 #1903 #1904 #1905 #1906 #1907 #1908 #1909 #1910 #1911 #1912 #1913 #1914 #1915 #1916 #1917 #1918 #1919 #1920 #1921 #1922 #1923 #1924 #1925 #1926 #1927 #1928 #1929 #1930 #1931 #1932 #1933 #1934 #1935 #1936 #1937 #1938 #1939 #1940 #1941 #1942 #1943 #1944 #1945 #1946 #1947 #1948 #1949 #1950 #1951 #1952 #1953 #1954 #1955 #1956 #1957 #1958 #1959 #1960 #1961 #1962 #1963 #1964 #1965 #1966 #1967 #1968 #1969 #1970 #1971 #1972 #1973 #1974 #1975 #1976 #1977 #1978 #1979 #1980 #1981 #1982 #1983 #1984 #1985 #1986 #1987 #1988 #1989 #1990 #1991 #1992 #1993 #1994 #1995 #1996 #1997 #1998 #1999 #2000 #2001 #2002 #2003 #2004 #2005 #2006 #2007 #2008 #2009 #2010 #2011 #2012 #2013 #2014 #2015 #2016 #2017 #2018 #2019 #2020 #2021 #2022 #2023 #2024 #2025 #2026 #2027 #2028 #2029 #2030 #2031 #2032 #2033 #2034 #2035 #2036 #2037 #2038 #2039 #2040 #2041 #2042 #2043 #2044 #2045 #2046 #2047 #2048 #2049 #2050 #2051 #2052 #2053 #2054 #2055 #2056 #2057 #2058 #2059 #2060 #2061 #2062 #2063 #2064 #2065 #2066 #2067 #2068 #2069 #2070 #2071 #2072 #2073 #2074 #2075 #2076 #2077 #2078 #2079 #2080 #2081 #2082 #2083 #2084 #2085 #2086 #2087 #2088 #2089 #2090 #2091 #2092 #2093 #2094 #2095 #2096 #2097 #2098 #2099 #2100 #2101 #2102 #2103 #2104 #2105 #2106 #2107 #2108 #2109 #2110 #2111 #2112 #2113 #2114 #2115 #2116 #2117 #2118 #2119 #2120 #2121 #2122 #2123 #2124 #2125 #2126 #2127 #2128 #2129 #2130 #2131 #2132 #2133 #2134 #2135 #2136 #2137 #2138 #2139 #2140 #2141 #2142 #2143 #2144 #2145 #2146 #2147 #2148 #2149 #2150 #2151 #2152 #2153 #2154 #2155 #2156 #2157 #2158 #2159 #2160 #2161 #2162 #2163 #2164 #2165 #2166 #2167 #2168 #2169 #2170 #2171 #2172 #2173 #2174 #2175 #2176 #2177 #2178 #2179 #2180 #2181 #2182 #2183 #2184 #2185 #2186 #2187 #2188 #2189 #2190 #2191 #2192 #2193 #2194 #2195 #2196 #2197 #2198 #2199 #2200 #2201 #2202 #2203 #2204 #2205 #2206 #2207 #2208 #2209 #2210 #2211 #2212 #2213 #2214 #2215 #2216 #2217 #2218 #2219 #2220 #2221 #2222 #2223 #2224 #2225 #2226 #2227 #2228 #2229 #2230 #2231 #2232 #2233 #2234 #2235 #2236 #2237 #2238 #2239 #2240 #2241 #2242 #2243 #2244 #2245 #2246 #2247 #2248 #2249 #2250 #2251 #2252 #2253 #2254 #2255 #2256 #2257 #2258 #2259 #2260 #2261 #2262 #2263 #2264 #2265 #2266 #2267 #2268 #2269 #2270 #2271 #2272 #2273 #2274 #2275 #2276 #2277 #2278 #2279 #2280 #2281 #2282 #2283 #2284 #2285 #2286 #2287 #2288 #2289 #2290 #2291 #2292 #2293 #2294 #2295 #2296 #2297 #2298 #2299 #2300 #2301 #2302 #2303 #2304 #2305 #2306 #2307 #2308 #2309 #2310 #2311 #2312 #2313 #2314 #2315 #2316 #2317 #2318 #2319 #2320 #2321 #2322 #2323 #2324 #2325 #2326 #2327 #2328 #2329 #2330 #2331 #2332 #2333 #2334 #2335 #2336 #2337 #2338 #2339 #2340 #2341 #2342 #2343 #2344 #2345 #2346 #2347 #2348 #2349 #2350 #2351 #2352 #2353 #2354 #2355 #2356 #2357 #2358 #2359 #2360 #2361 #2362 #2363 #2364 #2365 #2366 #2367 #2368 #2369 #2370 #2371 #2372 #2373 #2374 #2375 #2376 #2377 #2378 #2379 #2380 #2381 #2382 #2383 #2384 #2385 #2386 #2387 #2388 #2389 #2390 #2391 #2392 #2393 #2394 #2395 #2396 #2397 #2398 #2399 #2400 #2401 #2402 #2403 #2404 #2405 #2406 #2407 #2408 #2409 #2410 #2411 #2412 #2413 #2414 #2415 #2416 #2417 #2418
```

```

FullPort; ClearViewPort; SetLineStyle(0,0,1); qq:=35/6;
case sk of
  1 : i:=1;
  2 : i:=2;
  3 : i:=6;
  4,5,6 : i:=12;
end;
Kratka[2]:=(MaxY+1)/(3.5*i); Kratka[1]:=(MaxX+1)/(qq*i);
k:=Ord(Elkw[1]); if Ek1=0 then l:=1 else l:=Ek1;
if Ek2=0 then m:=1 else m:=Ek2; n:=Ord(c1[1]);
for i:=Ord(NN[1]) downto k0 do begin
  if i<>Ord(NN[1]) then begin Nkw:=30; k:=Ord('A'); l:=1; m:=1; n:=Ord('A') end;
  for j:=Nkw to k1 do begin
    repeat
      repeat
        repeat
          FullPort; ClearDevice;
          Wprow(i,j,k,l,m,n); lan1:=diag(sk); Zapis(lan1[1]); Inc(n);
        until n>k5;
        if i<>Ord(NN[1]) then begin Inc(m); n:=Ord(c1[1]) end
        else begin c11[1]:=chr(n); nael[1]:=chr(k); krok(na,nak,nael,l,m,c11);
          n:=Ord(c11[1]); k:=Ord(nael[1]) end;
        until m>k4;
        if i<>Ord(NN[1]) then begin Inc(l);
          if Ek2=0 then m:=i else m:=Ek2; n:=Ord(c1[1]) end;
        until l>k3;
        if i<>Ord(NN[1]) then begin Inc(k);
          if Ek1=0 then l:=1 else l:=Ek1; if Ek2=0 then m:=1 else m:=Ek2; n:=Ord(c1[1]) end;
        until k>k2;
        if i<>Ord(NN[1]) then begin
          k:=Ord(Elkw[1]); if Ek1=0 then l:=1 else l:=Ek1;
          if Ek2=0 then m:=1 else m:=Ek2; n:=Ord(c1[1]) end
        end;
        if i<>Ord(NN[1]) then begin
          k:=Ord(Elkw[1]); if Ek1=0 then l:=1 else l:=Ek1;
          if Ek2=0 then m:=1 else m:=Ek2; n:=Ord(c1[1]) end
        end;
      end;
    end;
  end;
end;
3 : begin
sk:=0; kole; oknoobw(1,1,80,24); kursor; otworzwy('a:\wspol.txt');
wyd:=Menu(3,wydr,''); przegladanie; f:=false; SetGraphMode(GetGraphMode);
if wyd in [1,2] then begin Godlo(NN,Nkw,Elkw,Ek1,Ek2,c1,k0,k1,k2,k3,k4,k5,false,1);
  Godlo(NN11,Nkw1,Elkw1,Ek11,Ek21,c11,k0,k1,k2,k3,k4,k5,false,2) end;
if length(NN)=0 then NN:=NarSorLewX; if length(NN11)=0 then NN11:=NarDolPrX;
RestoreCrtMode;
if not PrinterOk then begin
  case GraphDriver of
    CGA : TextMode(CB0);
    HercMono :TextMode(Mono)
  end;
  TextBackground(Blue); ClrScr;GotoXY(17,24); TextColor(Red); TextBackground(7);
  Write('Drukarka niedostepna lub niesprawna !!!');
  repeat
    Kom25(' Udostepnij drukarke i naciśnij dowolny klawisz ....',15,ch,fl);
  until PrinterOk end; Textbackground(blue); men:=0;
na:=NN; nak:=Nkw; nael:=Elkw; nael:=Ek1; nae2:=Ek2; naci:=c1; wsc:=false; sk:=0;
if c1<>' ' then naci:=c1 else naci:=' ';
repeat
  l1:=Lgodlo(Ord(na[1]),nak,Ord(nael[1]),nael,nae2,Ord(naci[1]),false)+' NUM';
  i:=4;
  while (i<9) and (l1[i]<>' ') do begin
    if l1[i]=' ' then l12:=Concat(copy(l1,1,i-1),'*.NUM'); Inc(i) end;
  Dir(l12,lista);
  if FileExist(l11) then begin
    men:=men+1; Assign(zb,l11); Reset(zb); wsc:=true; Czyt(wyd,men) end
  else begin ClrScr; GotoXY(20,13); Write(' B R A K Z B I D R U :',l11); delay(500); f:=true end;
  f:=false;
  if k2=0 then begin nael:=''; nael:=0; nae2:=0; naci:='' end;
  if k3=0 then begin nael:=0; nae2:=0; naci:='' end;
  if k4=0 then begin nae2:=0; naci:='' end;
  if k5=0 then begin nae2:=0; naci:='' end;

```

```
if koniec(na,nak,nael,nael,nae2,naci,NN1i,Nkw1,Elkw1,Ek1i,Ek2i,c1i,kol,true) then
  if not koniec(na,nak,nael,nael,nae2,naci,NN1i,Nkw1,Elkw1,Ek1i,Ek2i,c1i,wier,false) then begin
    na[i]:=chr(Ord(na[i])-1); nak:=Nkw; nael:=Elkw; nael:=Ek1; nae2:=Ek2; naci:=ci end
    else f:=true
  else krok(na,nak,nael,nael,nae2,naci)
until (wyd=1) or (men=80) or f;
if wsc then begin
  case wyd of
    1 : WydrukP;
    2 : WydrukS(men);
  end;
  zamknijwy; ClrScr end
else begin ClrScr; GotoXY(20,13); Write(' B R A K   D A N Y C H   W   T A B L I C Y '); delay(500) end;
end
else halt(99);
end;
wyb:=NN[i]=KON;
end end end.
```

```

unit bibliz;

interface

uses Crt,Dos,Turbo3,Graph,Pamiec,Kolbib;

type st3 =string[3];

procedure dzwiek;
  {generuje dzwiek}

procedure kursor;
  { Ustawia kursor w prawym dolnym rogu ekranu. }

procedure oknacbw(x1,y1,x2,y2:integer);
  { Czysci fragment ekranu zadany wspolrzednymi x,y .
  Rysuje podwojna obwiednie na skraju tego fragmentu.}

procedure czytczyn(var zn :char; var func :boolean);
  {Czyta z znak z klawiatury, a jesli nacisnieto klawisz funkcyjny
  o kodzie podwojnym to zn=drugi znak, func=true}

procedure pamietajekr(var pomekr:ekran);
  {zapamietuje zawartosc ekranu w tablicy pomekr}

procedure odtworzekr(pomekr:ekran);
  {odtworza zapamietany w tablicy pomekr obraz ekranu}

procedure liniegorne(n :integer; tekst :st78);
  {pisze tekst na ekranie w linii n - jesli n=2 kolor tekstu normalnego ,
  inaczej jest to tekst specjalny}

procedure karta;
  {zapewnienie zgodnosci programow z karta we/wy dla kart CGA i Hercules.
  Dla pozostalych kart komunikat i przerwanie programu}

procedure wrster(nazwa:NameStr;nrmenu,nrpoz,nrwyn,druk:integer);
  {Zapisuje na pliku zmienne sterujace przetwarzaniem w pakiecie MIKRO-INZ}

function rdster:boolean;
  {Czyta z pliku zmienne sterujace przetwarzaniem w pakiecie MIKRO-INZ
  sprawdzajac czy zadanie jest uruchamiane za posrednictwem menu}

procedure mwf;
  {Wyprowadzenie na ekran napisow identyfikujacych wykonawce (do zadan)}

procedure piszbyte(wart:byte;var lin:string;x,d:byte);

procedure piszinteger(wart:integer;dlug:byte;var lin:string;x,d:byte);

procedure piszreal(wart:real;dlug,poprzec:byte;var lin:string;x,d:byte);

procedure piszstring(s:string;var lin:string;x,d:byte);

procedure piszstrings(s:string;var lin:string;x,d:byte);

procedure strzalki(war,ofwrsz,ilwrsz,ilwrsznaekr,nrstr,ilstr,xpt,xkt,ypt:byte);
  { pisze strzalki pokazujace mozliwe kierunki przegladania
  tabeli, jesli nie miesci sie na jednym kadrze ekranu}

procedure piszdowlin(dlugs,pocz,kon,ye,druk:byte;lin:string;var wy:text);

procedure otworzwy(nazwa:string);

procedure zamknijwy;

procedure ds(var s:string;x,d:byte;co:string);

procedure dhl :

```

```

procedure di(var s:string;x,d,c:byte;co:integer);
procedure dr(var s:string;x,d,c,p:byte;co:real);
procedure przegladanie;
procedure dosr(nrlin :integer; txt : st27);
    { W linii nrlin pisze srodkowo na ekranie tekst do 27 znakow}
procedure bezobw(x1,y1,x2,y2 : integer);
    { Czysci fragment ekranu zadany wspolrzednymi x,y.}
procedure brakzbioru(txt : st27);
procedure czytstring(var st:string;x,y,dt:byte);
    { czytanie z klawiatury tekstu do dt znakow (dt<=20) }
procedure p(s,ss:string);
procedure swiec(poc : byte; prop : st3; k:byte);
procedure pisztxt(y:byte;txt:string);

```

Implementation

```

procedure dzwiek;
begin
    sound(523); delay(50); nosound
end;

procedure kursor;
begin
    gotoxy(80,24)
end;

procedure oknoobw(x1,y1,x2,y2:integer);
var i,i1,i2,k1,k2,x,y:integer;

    procedure pisz(var x:integer;y,nrzn:integer);
        var adr:integer;
        begin
            adr:=(x-1)+(y-1)*80; mem[baza:adr*2]:=nrzn; x:=x+1
        end;

begin
    {oknoobw}
    window(x1,y1,x2,y2); clrscr; window(1,1,80,25);
    i1:=x1+1; i2:=x2-1; k1:=y1+1; k2:=y2-1; x:=x1;
    pisz(x,y1,201); for i:=i1 to i2 do pisz(x,y1,205); pisz(x,y1,187);
    for i:=k1 to k2 do begin x:=x1;pisz(x,i,186) end;
    for i:=k1 to k2 do begin x:=x2;pisz(x,i,186) end;
    x:=x1; pisz(x,y2,200); for i:=i1 to i2 do pisz(x,y2,205); pisz(x,y2,188)
end;

procedure czytczyn(var zn :char; var func :boolean);
begin
    func:=false; zn:=ReadKey;
    if (zn=#27) and keypressed then begin read(kbd,zn); func:=true end
end;

procedure pamietajekr(var pomekr:ekran);
var i:integer;
begin
    for i:=0 to 3999 do pomekr[i]:=mem[baza:i]
end;

procedure odtworzekr(pomekr:ekran);
var i:integer;
begin
    for i:=0 to 3999 do mem[baza:i]:=pomekr[i];
end;

```

```

procedure liniegorne(n :integer; tekst :st78);
var i:integer;
begin
  if n=2 then kola else kole;
  gotoxy(2,n); for i:=1 to 78 do write(' '); gotoxy(2,n); write(tekst);
end;

procedure karta;
var i,j:integer;
begin
  detectgraph(i,j);
  if i=7 then baza:=#B000
  else if i<>1 then begin writeln;
    writeln('Oprogramowano karty CGA i Hercules'); halt end;
end;

procedure dosr(nrln :integer; txt : st27);
begin
  kolw; gotoxy(4i-length(txt) div 2,nrln); write(txt); cursor
end;

procedure brakzbioru(txt : st27);
  {sygnalizacja braku zbioru, przerwanie programu}
begin
  dzwiek; kolw; oknoobw(21,7,60,19); dosr(9,'NIEKOMPLETNY SYSTEM');
  dosr(10,'-----'); dosr(12,'Brak zbioru'); dosr(13,txt); halt
end;

procedure wrster(nazwa:NameStr;nrmenu,nrpoz,nrwyn,druk:integer);
type rekster=record
  naz:NameStr;
  nrm,nrp,nrw,dldr:integer
end;
var s:rekster;
  f:file of rekster;

begin
  assign(f,'c:\mwinz\menu\ster.rek');
  {$I-} reset(f);{$I+}
  if ioresult<>0 then brakzbioru('sterowania (ster.rek)');
  read(f,s); close(f);
  with s do begin
    if nazwa<>' then naz:=nazwa; if nrmenu=0 then nrm:=nrmenu;
    if nrpoz=0 then nrp:=nrpoz; if nrwyn<>0 then nrw:=nrwyn;
    if druk=0 then dldr:=druk end;
  rewrite(f); write(f,s); close(f)
end;

function rdster:boolean;
type rekster=record
  naz:NameStr;
  nrm,nrp,nrw,dldr:integer
end;
var s:rekster;
  f:file of rekster;

begin
  assign(f,'c:\mwinz\menu\ster.rek'); reset(f); read(f,s); close(f);
  rdster:={(s.nrm)=2} and {(s.nrm)<=17)}
end;

procedure mwf;
const f:array [1..14] of byte = (177,182,184,183,177,143,191,197,146,165,
  150,204,192,194);
  m:array [1..9] of byte = (190,188,192,197,196,164,194,197,211);
var i:integer;
begin
  kole; gotoxy(4,1);
  for i:=1 to 9 do write(chr(m[i]-111-i-(i mod 4))); gotoxy(64,1);
  for i:=1 to 14 do write(chr(f[i]-105-i-(i mod 3)));
end;

```

```

var i,p:byte;
    st,s:string;
begin
    st:=''; str(wart:d,s);
    p:=(d-length(s)) div 2;
    for i:=1 to p do st:=st+' '; st:=st+s;
    for i:=1 to d-length(s)-p do st:=st+' ';
    for i:=1 to d do lin[x+i-1]:=st[i]
end;

```

```

procedure piszinteger(wart:integer;dlug:byte;var lin:string;x,d:byte);
var i,p:byte;
    st,s:string;
begin
    st:=''; str(wart:dlug,s);
    p:=(d-length(s)) div 2;
    for i:=1 to p do st:=st+' '; st:=st+s;
    for i:=1 to d-length(s)-p do st:=st+' ';
    for i:=1 to d do lin[x-i+1]:=st[i]
end;

```

```

procedure piszreal(wart:real;dlug,pozzec:byte;var lin:string;x,d:byte);
var i,p:byte;
    st,s:string;
begin
    st:=''; str(wart:dlug;pozzec,s);
    if wart<1.0 then begin s:=copy(s,Pos('.',s),length(s)-Pos('.',s)+1);
        if wart=0.0 then s:=copy(s,1,2) end;
    i:=length(s);
    for p:=i downto 2 do if (s[i]='0') and (i>2) then Dec(i);
    if wart<1.0 then s:=copy(s,1,i); p:=(d-length(s)) div 2;
    for i:=1 to p do st:=st+' '; st:=st+s;
    for i:=1 to d-length(s)-p do st:=st+' ';
    for i:=1 to d do lin[x-1+i]:=st[i]
end;

```

```

procedure piszstring(s:string;var lin:string;x,d:byte);
var i:byte;
    st:string;
begin
    st:=s;
    for i:=length(s)+1 to d do st:=st+' ';
    for i:=1 to d do lin[x-i+1]:=st[i]
end;

```

```

procedure piszstrings(s:string;var lin:string;x,d:byte);
var i,p:byte;
    st:string;
begin
    st:=''; p:=(d-length(s)) div 2;
    for i:=1 to p do st:=st+' '; st:=st+s;
    for i:=1 to d-length(s)-p do st:=st+' ';
    for i:=1 to d do lin[x-1+i]:=st[i]
end;

```

```

procedure strzalki(war,ofwrsz,ilwrsz,ilwrsznaekr,nrstr,ilstr,xpt,xkt,ypt:byte);
    procedure piszstrz(st,kol,x,y:byte);
    begin
        mem[baza:(x-1+(y-1)*80)*2]:=st; mem[baza:(x-1+(y-1)*80)*2+1]:=kol;
    end;
begin
    if war=0 then piszstrz(ord(wer),le+16*te,2,ypt)
    else
        if ofwrsz<>0 then piszstrz(30,lest+16*te,2,ypt);
    if (ofwrsz<ilwrsz-ilwrsznaekr) then
        if war=1 then piszstrz(31,lest+16*te,2,23)
        else piszstrz(ord(wer),le+16*te,2,23)
    else
        if ofwrsz+ilwrsznaekr=ilwrsz then piszstrz(ord(wer),le+16*te,2,23);
    if war=0 then begin
        if xpt=2 then piszstrz(ord(nar[4]),le+16*te,2,ypt-1)

```

```

if nrstr<>ilstr then piszstrz(ord(nar[5]),le+16*te,xkt,ypt-1)
else piszstrz(ord(nar[6]),le+16*te,xkt,ypt-1) end
else begin
  if nrstr<>1 then piszstrz(17,lest+16*te,xpt,ypt-1);
  if nrstr<>ilstr then piszstrz(16,lest+16*te,xkt,ypt-1)
end;
end;

procedure pizsdowlin(dlugs,pocz,kon,ye,druk:byte;lin:string;var wy:text);
var i:byte;
begin
  if druk=0 then begin
    for i:=1 to dlugs do mem[baza:(i+(ye-1)*80)*2]:=ord(lin[i]);
    for i:=dlugs+1 to dlugs+kon-pocz+1 do
      mem[baza:(i+(ye-1)*80)*2]:=ord(lin[i-dlugs-1+pocz]);
    for i:=dlugs+2+kon-pocz to 78 do mem[baza:(i+(ye-1)*80)*2]:=32;
    for i:=1 to 78 do mem[baza:(i+(ye-1)*80)*2+1]:=le+16*te end
  else begin
    for i:=1 to dlugs do write(wy,lin[i]);
    for i:=dlugs+1 to dlugs+kon-pocz+1 do write(wy,lin[i-dlugs-1+pocz]);
    writeln(wy)
  end end;

procedure otworzwy(nazwa:string);
begin
  assign(wy,nazwa); rewrite(wy)
end;

procedure zamknijwy;
begin
  close(wy)
end;

procedure ds(var s:string;x,d:byte;co:string);
var i:byte;
begin
  for i:=x to x+d-1 do
    if i-x+1>length(co) then s[i]:=' ' else s[i]:=co[i-x+1]
end;

procedure db(var s:string;x,d,c,co:byte);
var st:string;
begin
  str(co:c,st); ds(s,x,d,st)
end;

procedure di(var s:string;x,d,c:byte;co:integer);
var st:string;
begin
  str(co:c,st); ds(s,x,d,st)
end;

procedure dr(var s:string;x,d,c,p:byte;co:real);
var st:string;
begin
  str(co:c,p,st); ds(s,x,d,st)
end;

procedure swiec(poc : byte; prop : st3; k:byte);
begin
  if k=0 then kolw else kolot;
  gotoxy(poc,15); write(' ',prop,' '); kursor
end;

procedure pisatxt(y:byte;txt:string);
begin
  if txt<>' ' then begin gotoxy(40-(length(txt) div 2),y); write(' ',txt,' ')
  end
end;

procedure przegladanie;
const prop:piszstrz(16,21 of string(31)-TAK: NIE);

```

```

var i:byte;
enter:boolean;
pocz:array [1..4] of byte;
pomekr:ekran;

procedure objas(st:string);
var i,p:integer;
    s:string;
begin
    p:=(80-length(st)) div 2; s:='';
    for i:=1 to p do s:=s+' '; s:=s+st;
    for i:=length(s)+1 to 80 do s:=s+' ';
    for i:=1 to 80 do begin
        mem[baza:(i+1919)*2]:=ord(s[i]); mem[baza:(i+1919)*2+1]:=lobj+16*tobj;
    end;
end;

begin
    pamietajekr(pomekr); kolw;
    bezobw(25,9,57,17); oknoobw(28,10,54,16);
    pisztxt(10,'Przeглядanie tabel');
    pisztxt(12,'Czy chcesz przegladac');
    pisztxt(13,'tabulogramy wynikowe?');
    pocz[1]:=36; pocz[2]:=41; swiec(pocz[1],propozycja[1],1); swiec(pocz[2],propozycja[2],0);
    i:=1; enter:=false; objas('Wybierz odpowiedz');
    while not enter do begin
        czytczyn(zn,func);
        if func then
            case zn of
                #75:begin
                    swiec(pocz[i],propozycja[i],0); if i=1 then i:=2 else i:=i; swiec(pocz[i],propozycja[i],1)
                end;
                #77:begin
                    swiec(pocz[i],propozycja[i],0); if i=2 then i:=1 else i:=2; swiec(pocz[i],propozycja[i],1)
                end end
            else
                case zn of
                    #13:begin enter:=true; olo:=(i=1) end;
                    #27:enter:=true;
                end end;
        kursor; odtworzekr(pomekr)
    end;

procedure bezobw(x1,y1,x2,y2 :integer);
begin
    window(x1,y1,x2,y2); clrscr; window(1,1,80,25)
end;

procedure p(s,ss:string);
begin
    kolobk; write(s); kolobt; write(ss)
end;

procedure czytstring(var st:string;x,y,d:byte);
var l:string;
    i,j,k:byte;
    enter:boolean;

procedure poslug;
{ opis funkcji klawiszy do wprowadzanie tekstu z klawiatury }
begin
    gotoxy(1,25);
    p('znak','ZNAK TEKSTU'); p(#27+#26,'PRZESUW KURSODR'); p(#27+'(bs) Del','KAS.ZNAKU');
    p('Enter','KONIEC WPROW');p('Esc','REZYGNACJA');
    mem[baza:3999]:=lobt+16*tobt; gotoxy(80,24)
end;

function czysckon(l:string):string;
var i:byte;
    kon:boolean;
begin
    if l[1]=' ' then czysckon:=l[2..length(l)] else

```

```

kon:=false;
for i:=dt-1 downto 0 do
  if (not kon) and (l[i]<>' ') then begin
    kon:=true; czysckon:=copy(l,i,i)
  end;
end
end;

begin
if (not func) and ((zn>=' ') or (zn=#13)) then begin
  poslug; enter:=false; l:='';
  for i:=1 to dt do l:=l+' ';
  for i:=1 to length(st) do l[i]:=st[i];
  kolot; gotoxy(x,y);
  if zn<>#13 then begin
    l[1]:=zn; write(l); gotoxy(x+1,y); i:=2 end
  else i:=1;
  while not enter do begin
    gotoxy(x,y); write(l); gotoxy(x+1-1,y); czytczyn(zn,func);
    if func then
      case zn of
        #75:if i<>1 then i:=i-1;
        #77:if i<>dt then i:=i+1;
        #71:i:=1;
        #79:begin
          for j:=1 to dt do if l[j]<>' ' then k:=j;
            if k<>dt then k:=k+1; i:=k;
          end;
        #83:begin
          for j:=i+1 to dt do l[j-1]:=l[j]; l[dt]:=' ';
          end;
        else dzwiek
      end
    end
  else
    case zn of
      #8:if i<>1 then begin for j:=i to dt do l[j-1]:=l[j];
        l[dt]:=' '; i:=i-1;
      end;
      #27:begin
        enter:=true; gotoxy(x,y); write(st);
        for i:=length(st)+1 to dt do write(' ');
      end;
      #13:begin
        st:=l; enter:=true;
      end;
    else begin
      for j:=dt-1 downto i do l[j+1]:=l[j]; l[i]:=zn;
      if i<>dt then i:=i+1 else begin
        dzwiek; st:=l; gotoxy(x,y); write(l); enter:=true;
      end
    end
  end;
end;
  kursor end
else dzwiek
end;
end.

```

```

procedure wr(l:integer; s :st78);
  (* W linii l pisze srodkowo tekst do 78 znakow. *)
  begin gotoxy(41-length(s) div 2,l);
    write(s)
  end;

procedure czolowka;
  procedure pierwszykadr;
    (* Wyswietla na ekranie czolowke zadania *)
    type st78=string[78];
    var zn:char;
        pp:string[20];
        i :byte;
        f :boolean;

    procedure podkresi(y:integer);
      (* W linii y rysuje podkreslenie. *)
      var i:integer;
      begin
        gotoxy(3,y);
        for i:=3 to 78 do write(chr(196))
      end;

  begin
    {czolowka}
    oknoobw(1,1,80,24);
    wr(2,'A K A D E M I A   O B R O N Y   N A R O D O W E J'); podkresi(3);
    wr(6,'Komputerowa mapa wspolczynnikow');
    WR(11,'S R O D O W I S K O'); wr(15,''); gotoxy(41,17);
    write('AUTORZY :'); GotoXY(40,18); FillChar(pp,sizeOf(pp),196);
    for i:=1 to 20 do write(pp[i]); gotoxy(36,19);
    write('projekt koncepcyjny');gotoXY(37,20);
    write('oraz zlozenia.   pplk dr J.Stepak'); GotoXY(36,21);
    write('oprogramowanie   pplk mgr inz S.Dabrowski'); podkresi(22);
    wr(23,'WARSZAWA           GRUDZIEN           1989');
    GotoXY(25,1); TextColor(15); zn:=ReadKey;
  end;

  procedure naglowek;
    {**** Naglowek zadania ****}
    var i:integer;
    begin
      kolo; clrscr; oknoobw(1,2,80,24);
      liniegorne(1,'           S R O D O W I S K O');
      wr(8,'W P R O W A D Z E N I E'); wr(13,'A K T U A L I Z A C J A');
      wr(18,'O B L I C Z A N I E'); cursor; delay(1000)
    end;

  begin {przygotowanie}
    (* sprawdzenie karty we/wy i wczytanie zestawu kolorow *)
    karta; czytkol; pierwszykadr; naglowek
  end;

```

```

unit Inne;

INTERFACE

uses Dos,Graph,Crt,Model,printer,Dodatek;

type st2 = string[2];
    tab2= array [1..3,1..3] of real;

procedure Inicjacja(var GraphDriver,GraphMode :integer);

procedure Ekran(i :byte; x1,y1,x2,y2 :integer; f :boolean; var p :pointer);

procedure Ramka(i,ck,w :byte);

procedure dtext(x,y,ilzn :integer; var st :string; var fl :boolean; ii : integer);

function Liczba(z1,z2,p,r,q :integer; var l :boolean):integer;

function LiczbaR(z1,z2 :real; p,r,q :integer; wsk :real; var l,li :boolean):real;

procedure KomunikatG(a :string; i :integer; var sx :real; var ch :char);

IMPLEMENTATION

procedure Inicjacja(var GraphDriver,GraphMode :integer);
    var c :real;
        ErrorCode : integer;
        OldExitProc : pointer;

    {#F+}
    procedure MyExitProc;
        begin
            ExitProc := OldExitProc; CloseGraph
        end;
    {#F-}

    begin
        DirectVideo:=False; OldExitProc:=ExitProc; { save previous exit proc}
        ExitProc:=@MyExitProc; { insert our exit proc in chain}
        GraphDriver:=Detect; { use autodetection}
        InitGraph(GraphDriver, GraphMode, ''); { activate graphics}
        ErrorCode:=GraphResult; { error?}
        if ErrorCode<>grOk then begin
            Writeln('Blad trybu graficznego nr : ', GraphErrorMsg(ErrorCode)); Halt(1)
        end; c:=99.99;
        lit[1]:=TextWidth('M'); lit[2]:=TextHeight('M'); MaxX:=GetMaxX; MaxY:=GetMaxY;
        case GraphDriver of
            CGA : begin
                if c=0.0 then GraphMode:=CGAC2 else GraphMode:=CGAHi;
                Kolor:=14; BkKolor:=10; Kolor1:=0; BkKolor1:=9; BkKolor2:=0; Kolor2:=15;
                q:=0.41
            end;
            MCGA : if c=0.0 then GraphMode:=MCGAC2 else GraphMode:=MCGAHi;
            EGA : GraphMode := EGAHi;
            EGA64 : GraphMode := EGA64Hi;
            HercMono : begin
                Kolor:=15; BkKolor:=0; Kolor1:=7; BkKolor1:=0; BkKolor2:=0; Kolor2:=15;
                q:=0.6922
            end;
            EGAMono : ;
            FC3270 : GraphMode:=FC3270Hi;
            ATT400 : if c=0.0 then GraphMode:=ATT400C2 else GraphMode:=ATT400Hi;
            VGA : GraphMode:=VGAHi;
        end;
        SetGraphMode(GraphMode); MaxColor:=GetMaxColor; MaxX:=GetMaxX; MaxY:=GetMaxY
    end;

procedure Ekran;

```

```

case i of
  1 : begin GetMem(p,ImageSize(x1,y1,x2,y2)); GetImage(x1,y1,x2,y2,p^) end;
  2 : begin SetGraphMode(GetGraphMode); FullPort; ClearDevice; PutImage(x1,y1,p^,NormalPut) end;
  3 : begin FullPort; PutImage(x1,y1,p^,NormalPut) end end;
if f then FreeMem(p,ImageSize(x1,y1,x2,y2)); end;

procedure Ramka;
var l,li :integer;
    iinf :ViewportType;
begin
  GetViewSettings(iinf); with iinf do SetViewport(x1,y1,x2,y2,ClipOn);
  if w=0 then SetColor(w); with iinf do Rectangle(0,0,x2-x1,y2-y1); SetColor(1)
end;

procedure dtext(x,y,ilzn :integer; var st :string; var fl :boolean; ii : integer);
var pom :char;
    l :string[16];
    i,j :integer;
    enter,sf,p :boolean;
    inf,inf1 :ViewportType;
begin
  enter:=false; FillChar(l,16,' '); i:=1; sf:=Length(st)=0; ps:=true; l:='';
  if not fl then begin GetViewSettings(inf); FullPort;
  with inf do SetViewport(x1+x,y1+y,x1+x+lit[1]*ilzn+2,y1+y+lit[2],ClipOn);
  ClearViewport; if not sf then OutTextXY(2,0,st) end;
  while not enter do begin
    if fl then GotoXY(x+i-1,y); pom:=ReadKey;
    if p and not fl and (i=1) and not (pom in [KON,ESC,ENT]) then begin st:=' '; sf:=true;
    p:=false; ClearViewport; FullPort; with inf do SetViewport(x1,y1,x2,y2,ClipOn) end;
    if sf then begin
      case pom of
        #0 : begin
          pom:=ReadKey; if pom=#79 then begin l:=l+#0+#79; enter:=true; fl:=true end;
          end;
        #8 : begin
          if i<>1 then begin Dec(i); l:=copy(l,i,i);
          if fl then begin Gotoxy(x+i-1,y); Write(' '); Gotoxy(x+i-1,y) end
          else begin SetColor(0); OutTextXY(x+(i-1)*lit[1]+1,y,l[i]); SetColor(1); end;
          end end;
        #13 : begin enter:=true; fl:=i=1; if i=1 then l[i]:=ENT else Dec(i) end;
        #27 : begin l[i]:=ESC; enter:=true; fl:=true end;
        'w','W': begin enter:=true; l:=pom end;
        else if i<>ilzn+1 then begin if i=1 then pom:=UpCase(pom);
          if fl then write(pom) else OutTextXY(x+(i-1)*lit[1]+1,y,pom);
          l:=Concat(l,pom); Inc(i) end
        end end
      else begin enter:=true; fl:=pom=ESC; l:=Concat(l,pom) end;
    end; st:=copy(l,1,Length(l));
    with inf do SetViewport(x1,y1,x2,y2,ClipOn)
  end;

function LiczbaR(z1,z2 :real; p,r,q :integer; wsk :real; var l,li :boolean):real;
var a :real;
    b :integer;
    aa :string;
    inf :ViewportType;
    f,fl,ff:boolean;
begin
  a:=0.0;
  while a=0.0 do begin str(wsk:6:4,aa); ff:=l;
  if not l then GetViewSettings(inf) else Wbl(7,0,p,r,q,false,l); dtext(p,r,q,aa,ff,0);
  f:=aa[length(aa)]=KON; fl:=aa[length(aa)]=ESC; l:=f; li:=fl;
  if UpCase(aa[1])='W' then begin liczbaR:=99.99; EXIT end;
  if ((aa[1]=#0) and (aa[2]=KON)) or (aa[1]=ESC) then begin liczbaR:=wsk; exit end;
  if (aa[1]=#0) and (aa[2] in [KON,ESC]) then exit;
  if (aa[1]<>KON) then begin b:=0;
  if aa[1]=ENT then a:=wsk else begin
    if f or fl then aa:=copy(aa,1,length(aa)-1); if aa[length(aa)]='.' then aa:=Concat(aa,'0');
    if aa[1]='.' then aa:='0'+aa; Val(aa,a,b) end;
  if (a<z1) or (a>z2) or (b<>0) then begin
    if not l then GetViewSettings(inf);

```

```

    if not l then with inf do SetViewPort(x1,y1,x2,y2,Clip); a:=0 end
  else begin
    if l then begin TextBackground(0); TextColor(7); GotoXY(p,r); Write(a;q) end
    else begin with inf do SetViewPort(x1,y1,x2-2,y2-2,Clip); ClearViewPort; Str(a1;q-1,aa); Outtext(aa) end;
    LiczbaR:=a; a:=999
  end end
  else begin LiczbaR:=wsk; a:=999 end
end; l:=f or ff; ll:=f1 or ff; if not l then begin FullPort; with inf do SetViewPort(x1,y1,x2,y2,ClipOn) end
end;

```

```

function Liczba;
var a,b :integer;
    aa :string;
    inf :ViewPortType;
    f :boolean;
begin
  a:=0;
  while a=0 do begin
    if not l then GetViewSettings(inf); aa:=''; f:=1;
    if l then Wbl(7,0,p,r,q,false,l); dtext(p,r,q,aa,l,0); l:=f; f:=aa[length(aa)]=KON;
    if f then aa:=copy(aa,1,length(aa)-2); Val(aa,a,b);
    if (a<z1) or (a>z2) or (b<>0) then begin
      Wbl(7,4+128,p,r,q,true,l); if l then Wbl(7,0,p,r,q,false,l);
      if not l then with inf do SetViewPort(x1,y1,x2,y2,Clip); a:=0 end
    else begin
      if l then begin TextBackground(0); TextColor(7); GotoXY(p,r) end;
      if l then Write(a;q); Liczba:=a; a:=999
    end end; l:=f; if not l then begin FullPort; with inf do SetViewPort(x1,y1,x2,y2,ClipOn) end;
  end;
end;

```

```

procedure KomunikatG;
var lo,ll :integer;
    ps :pointer;
    c :string;
    y :integer;
    f :boolean;
begin
  if sx=99.99 then y:=0 else y:=20;
  ll:=Length(a)*lit[1]; lo:=(MaxX-ll-y*lit[1]) div 2; ll:=lo+ll;
  DknoKom(lo,MaxY-lit[2]-2,ll+2,MaxY,a);
  SetViewPort(ll+3,MaxY-lit[2]-2,ll+lit[1]*13,MaxY,ClipOn);
  Ramka(0,ll*lit[1],1);
  Case i of
    1 : ;
    2 : ;
    3 : begin
      repeat
        c:='TAK'; f:=false;
        dtext(3,1,3,c,f,1);
        until c[1] in ['T','N',#13];
        ch:=c[1] end;
    4 : begin
      repeat dtext(3,1,3,c,false) until UpCase(c[1]) in ['a','b','c','d',' '];
        ch:=UpCase(c[1]) end;
  end; SetViewPort(0,MaxY-lit[2]-2,MaxX,MaxY,ClipOn); ClearViewPort; end;
end.

```

```

unit #arbibi;

interface

uses Crt,Dos,Turbo3,Graph,Pamiec,Kolbib,Biblinz;

procedure objas(puste:integer;st:st78);
  { Wyszwietla w ostatniej linii ekranu linie informacyjna.
  Jesli puste=0 wtedy linia ta jest czysta w kolorze tla ekranu.
  Ustawia kursor w prawym dolnym rogu ekranu.}
procedure doopisu(st:st50);
  {wpisuje tekst do naglowka menu}
procedure readster(var nazwa:NameStr;var nrmenu,nrpoz,nrwn,druk:integer);
  {Czyta z pliku zmienne sterujace przetwarzaniem w pakiecie MIKRO-INZ}
function rdstere:boolean;
  {Czyta z pliku zmienne sterujace przetwarzaniem w pakiecie MIKRO-INZ
  sprawdzajac czy edytor jest uruchamiany za posrednictwem menu}

procedure czytstring5(var st:st5;x,y,dt:byte);

procedure czytstring10(var st:st10;x,y,dt:byte);

procedure czolowka(par:st78);

implementation

procedure objas(puste:integer;st:st78);
  var i:integer;
  begin
    if puste=0 then kole else kolobj;
    gotoxy(2,25); for i:=2 to 79 do write(' ');
    if puste<>0 then begin gotoxy(41-length(st) div 2,25); write(st) end;
    gotoXY(79,25)
  end;

procedure doopisu;
var i:integer;
    jest:boolean;
begin
  i:=length(st); jest:=false;
  while (not jest) or (i=0) do
    if (st[i]<>' ') then jest:=true else i:=i-1;
  if i=0 then st:='' else if i<>length(st) then delete(st,i+1,50);
  kole; gotoxy(16,2);
  for i:=1 to 50 do write(' '); gotoxy(41-(length(st) div 2),2); write(st)
end;

procedure readster;
type rekster=record
  naz:NameStr;
  nrn,nrp,nrw,dldr:integer
end;
var s:rekster;
    f:file of rekster;
begin
  assign(f,'c:\mwinz\menu\ster.rek');
  {#I-} reset(f);{#I+}
  if ioresult<>0 then brakzbioru('sterowania (ster.rek)');
  read(f,s); close(f);
  with s do begin
    nazwa:=naz; nrmenu:=nrn; nrpoz:=nrp; nrwn:=nrw; druk:=dldr
  end end;

function rdstere:boolean;
type rekster=record
  naz:NameStr;
  nrn,nrp,nrw,dldr:integer
end;
var s:rekster;
    f:file of rekster;

```

```

begin
  assign(f,'c:\mwinz\menu\ster.rek');
  {$I-} reset(f);{$I+}
  if ioreult=0 then begin
    read(f,s); close(f); rdstere:=((s.nrm=0) and (s.nrp=2)) end
  else rdstere:=false
end;

procedure czytstring5(var st:st5;x,y,dt:byte);
var s:string;
    i:byte;
begin
  s:=''; for i:=1 to length(st) do s:=s+st[i]; czytstring(s,x,y,dt);
  st:=''; for i:=1 to length(s) do st:=st+s[i];
end;

procedure czytstring10(var st:st10;x,y,dt:byte);
var s:string;
    i:byte;
begin
  zn:=#13; func:=false; s:='';
  for i:=1 to length(st) do s:=s+st[i]; czytstring(s,x,y,dt); st:='';
  for i:=1 to length(s) do st:=st+s[i];
end;

procedure wr(l:integer; s :st78);
  (**** W linii l pisze srodkowo tekst do 78 znakow. ****)
  begin gotoxy(41-length(s) div 2,l); write(s)
  end;

procedure czolowka;
  procedure pierwszykadr;
  (**** Wyświetla na ekranie czolowke zadania tj. ekran informacyjny. ****)
  var zn:char;
      pp:string[20];
      i :byte;

  procedure podkresl(y:integer);
  (**** W linii y rysuje podkreslenie. ****)
  var i:integer;
  begin
    gotoxy(3,y); for i:=3 to 78 do write(chr(196))
  end;

begin (czolowka)
  kolo; oknoobw(1,1,80,24);
  wr(2,'A K A D E M I A   O B R O N Y   N A R O D O W E J');
  podkresl(3); wr(6,'Komputerowa mapa współczynnika');
  WR(11,"S R O D O W I S K O"); wr(15,''); gotoxy(41,17);
  write('AUTORZY :'); GotoXY(40,18); FillChar(pp,sizeOf(pp),196);
  for i:=1 to 20 do write(pp[i]); gotoxy(40,19);
  write('projekt koncepcyjny');gotoXY(41,20);
  write('oraz zlozenia   ppik dr J.Stepak'); GotoXY(40,21);
  write('oprogramowanie   ppik mgr S.Dabrowski'); podkresl(22);
  wr(23,'WARSZAWA          GRUDZIEN          1989'); zn:=Readkey;
end;

procedure naglowek;
  (**** Naglowek zadania ****)
  var i:integer;
  begin
    kolo; clrscr;
    oknoobw(1,2,80,24); liniegorne(1,'S R O D O W I S K O');
    wr(8,'W P R O W A D Z E N I E'); wr(13,'A K T U A L I Z A C J A');
    wr(18,'O B L I C Z A N I E'); kursor; delay(1000)
  end;

begin (przygotowanie)
  (**** sprawdzenie karty wv/hv i wczytanie zestawu kolorow ****)
  czytkol; pierwszykadr; naglowek
end;

```

```

end.

unit udeklddal;

interface

uses Dos,Crt,Graph,Pamiec,Kolbib,Siblinz,Uprocedury;

procedure czytstring5(var st:st5;x,y,dt:byte);

procedure czytstring10(var st:st10;x,y,dt:byte);

procedure czolowka(par:st78);

implementation

procedure czytstring5(var st:st5;x,y,dt:byte);
var s:string;
    i:byte;
begin
s:=''; for i:=1 to length(st) do s:=s+st[i]; czytstring(s,x,y,dt);
st:=''; for i:=1 to length(s) do st:=st+s[i];
end;

procedure czytstring10(var st:st10;x,y,dt:byte);
var s:string;
    i:byte;
begin
zn:=#13; func:=false; s:='';
for i:=1 to length(st) do s:=s+st[i]; czytstring(s,x,y,dt); st:='';
for i:=1 to length(s) do st:=st+s[i];
end;

procedure wr(l:integer; s :st78);
(**** W linii l pisze srodkowo tekst do 78 znakow. ****)
begin gotoxy(41-length(s) div 2,l); write(s)
end;

procedure czolowka;
procedure pierwszykadr;
(**** Wyświetla na ekranie czolowke zadania tj. ekran informacyjny. ****)
var zn:char;
    pp:string[20];
    i :byte;

procedure podkresl(y:integer);
(**** W linii y rysuje podkreslenie. ****)
var i:integer;
begin
gotoxy(3,y); for i:=3 to 78 do write(chr(196))
end;

begin (czolowka)
kole; oknoobw(1,1,80,24);
wr(2,'A K A D E M I A O B R O N Y N A R O D D O W E J');
podkresl(3); wr(6,'Komputerowa mapa wspolczynnikow');
WR(11,'S R O D D O W I S K O'); wr(15,''); gotoxy(41,17);
write('AUTORZY : '); GotoXY(40,18); FillChar(pp,sizeOf(pp),196);
for i:=1 to 20 do write(pp[i]); gotoxy(40,19);
write('projekt koncepcyjny');gotoXY(41,20);
write('oraz zlozenia ppik dr J.Stepak'); GotoXY(40,21);
write('oprogramowanie ppik mgr S.Dabrowski'); podkresl(22);
wr(23,'WARSZAWA GRUDZIEN 1989'); zn:=Readkey;
end;

procedure naglowek;
(**** Naglowek zadania ****)
var i:integer;
begin
kole; clrscr;
oknoobw(1,2,80,24); liniegorne(1,'S R O D D O W I S K O');

```

```
wr(18,'D B L I C Z A N I E'); kursor; delay(1000)
end;
```

```
begin {przygotowanie}
  (**** sprawdzienie karty we/wy i wczytanie zestawu kolorow ****)
  czytkol; pierwszykadr; naglowek
end;
```

```
end.
```

```
unit uprocedury;
```

```
interface
```

```
uses Crt,Dos,Graph,Turbo3,Pamiec,Kolbib,Biblinz,Marbibl;
```

```
const znakikonca:set of char=[#72,#73,#75,#77,#80,#81];
```

```
konniefunc:set of char=[#27,#13];
```

```
poczatekliczby:set of char=['0','1','2','3','4','5','6','7','8','9'];
```

```
type a4st=array [1..4] of string;
```

```
tekran=^ekran;
```

```
procedure piszlongint(wart:longint;dlug:byte;var lin:string;x,d:byte);
```

```
procedure piszword(wart:word;dlug:byte;var lin:string;x,d:byte);
```

```
procedure piszshortint(wart:shortint;dlug:byte;var lin:string;x,d:byte);
```

```
procedure ostrzez(st:string);
```

```
{ Wyswietla w ostatniej linii ekranu linie informacyjna.
```

```
  Jesli puste=0 wtedy linia ta jest czysta w kolorze tla ekranu.
```

```
  Ustawia kursor w prawym dolnym rogu ekranu. }
```

```
procedure opisklaw(obl,obl:opklaw);
```

```
procedure swiatlo(x,y,d,kol:byte);
```

```
{podswietla kolorem okreslonym przez kol biezacy element tabeli
```

```
  kol=0 - brak podswietl.;
```

```
  =1 - wyroznienie;
```

```
  =2 - blad }
```

```
procedure swiatkol(x,yptn,yktn,d,kol:byte);
```

```
{ poswietla naglowek kolumny z biez. wartoscia
```

```
  kol=0 - brak podswietl.;
```

```
  =1 - podswietl }
```

```
procedure swiatwrsz(x,y,d,kol:byte);
```

```
{ poswietla naglowek wiersza z biez. wartoscia
```

```
  kol=0 - brak podswietl.;
```

```
  =1 - podswietl }
```

```
procedure zakresy(zakres:sti2;czyzakr,x,yptm,d:byte);
```

```
{ podswietla kolorem okreslonym przez kol biezacy element tabeli
```

```
  kol=0 - brak podswietl.;
```

```
  =1 - wyroznienie;
```

```
  =2 - blad }
```

```
procedure czytstringi6(var st:sti6;x,y,d:byte);
```

```
procedure poslug;
```

```
{ opis funkcji klawiszy do wprowadzania wart. z klawiatury }
```

```
procedure czytreal(var wart:real;dlicz,poprzec:byte;zakres1,zakres2:real;
```

```
  x,y,d:byte;var bylo:boolean);
```

```
{ czyta podana z klawiatury liczbe rzeczywista wart o dlugosci dlicz i poprzec miejscach po przecinku,
```

```
  badajac jej poprawnosc syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
```

```
  sygnalizuje blad, jesli liczba niepoprawna }
```

```
procedure czytword(var wart:word;dlicz,zakres1,zakres2:word;x,y,d:byte;
```

```
  var bylo:boolean);
```

```
{ czyta podana z klawiatury liczbe calkowita wart badajac jej poprawnosc
```

```
  syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
```

```
  sygnalizuje blad, jesli liczba niepoprawna }
```

```
procedure czytbyte(var wart:byte;dlicz,zakres1,zakres2,x,y,d:byte;
```

```
  var bylo:boolean);
```

```
{ czyta podana z klawiatury liczbe calkowita wart badajac jej poprawnosc
```

```
  syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
```

```
  sygnalizuje blad, jesli liczba niepoprawna }
```

```
procedure czytlongint(var wart:longint;dlicz:byte;
```

```
  zakres1,zakres2:longint;x,y,d:byte;var bylo:boolean);
```

```
{ czyta podana z klawiatury liczbe calkowita wart badajac jej poprawnosc
```

```
  syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
```

```
  sygnalizuje blad, jesli liczba niepoprawna }
```

```
procedure czytinteger(var wart:integer;dlicz:byte;
```

```
  zakres1,zakres2:integer;x,y,d:byte;var bylo:boolean);
```

```
{ czyta podana z klawiatury liczbe calkowita wart badajac jej poprawnosc
```

```
  syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
```

```
  sygnalizuje blad, jesli liczba niepoprawna }
```

```
procedure czytshortint(var wart:shortint;dlicz:byte;
```

```
  zakres1,zakres2:shortint;x,y,d:byte;var bylo:boolean);
```

```

syntaktyczna i sprawdzajac czy zawiera sie w zakresie zakres1-zakres2 ;
sygnalizuje blad, jesli liczba niepoprawna )
procedure czytdyskretnb(var wart:byte;od,do:byte);
procedure czytdyskretni(var wart:integer;od,do:integer);
procedure przewin(x1,y1,x2,y2,kier:byte);
  { wiersze ekranu od x1,y1 do x2,y2 sa przesuwane o 1 wiersz
  -1 w gore 1 w dol }
function oknopytania(illin:byte;txtg,txt1,txt2,txt3,txt4,txt5,txt6,txt7:string;
  propozycja:a4st;ilosc:byte):byte;
procedure oknosrodk(illin:byte;txtg,txt1,txt2,txt3,txt4,txt5,txt6,txt7:string);
procedure pizspacje(y:byte);
function raportnaekranie:byte;

```

implementation -

```

procedure pizlongint(wart:longint;dlug:byte;var lin:string;x,d:byte);
var i,p:byte;
    st,s:string;
begin
  st:=''; str(wart:dlug,s); p:=(d-length(s)) div 2;
  for i:=1 to p do st:=st+' '; st:=st+s;
  for i:=1 to d-length(s)-p do st:=st+' ';
  for i:=1 to d do lin[x+i-1]:=st[i]
end;

procedure pizsword(wart:word;dlug:byte;var lin:string;x,d:byte);
begin
  pizlongint(wart,dlug,lin,x,d);
end;

procedure pizshortint(wart:shortint;dlug:byte;var lin:string;x,d:byte);
begin
  pizlongint(wart,dlug,lin,x,d);
end;

procedure ostrzez(st:string);
var i,p:integer;
    s:string;
    ks,ps:array [1..80] of byte;
    zn:char;
    func:boolean;
begin
  for i:=1 to 80 do begin
    ps[i]:=mem[baza:(i+1919)*2]; ks[i]:=mem[baza:(i+1919)*2+1];
  end;
  p:=(80-length(st)) div 2; s:=''; for i:=1 to p do s:=st+' '; s:=s+st;
  for i:=length(s)+1 to 80 do s:=s+' ';
  for i:=1 to 80 do begin
    mem[baza:(i+1919)*2]:=ord(s[i]); mem[baza:(i+1919)*2+1]:=lobj+i6#tobj;
  end; dzwiek; czytczyn(zn,func);
  for i:=1 to 80 do begin
    mem[baza:(i+1919)*2]:=ps[i]; mem[baza:(i+1919)*2+1]:=ks[i];
  end end;

procedure opisklaw(obk,obt:opklaw);
var d,i,j,k,il:byte;
begin
  d:=0;
  for i:=1 to 10 do begin
    if length(obk[i])+length(obt[i])<>0 then il:=i;
    d:=d+length(obk[i])+length(obt[i]);
  end;
  if d<80 then begin
    for i:=1 to ((80-d) div 2) do obk[i]:=obk[i]+' ';
    for i:=1 to 80-d-((80-d) div 2) do obt[i]:=obt[i]+' ';
  end; k:=0;
  for i:=1 to il do begin
    for j:=1 to length(obk[i]) do begin
      if ord(obk[i,j])>200 then mem[baza:(k+1920)*2]:=ord(obk[i,j])-200
      else mem[baza:(k+1920)*2]:=ord(obk[i,j]);
      mem[baza:(k+1920)*2+1]:=lobk+i6#tobj; k:=k+1;
    end;
  end;
end;

```

```

for j:=1 to length(obt[i]) do begin
  mem[baza:(k+1920)*2]:=ord(obt[i,j]);
  mem[baza:(k+1920)*2+1]:=lobt+16*tobt; k:=k+1;
end end end;

```

```

procedure swiatlo(x,y,d,kol:byte);
var i,tlo,litery:byte;
begin
  case kol of
    0:begin tlo:=te;litery:=le end;
    1:begin tlo:=tot;litery:=lot end;
    2:begin tlo:=tbl;litery:=lbl end;
  end;
  for i:=x-1 to x+2*d do mem[baza:l+2*(i+(y-1)*80)]:=litery+16*tlo;
end;

```

```

procedure swiatkol(x,yptn,yktn,d,kol:byte);
var i,j,tlo,litery:byte;
begin
  if kol=0 then begin tlo:=te;litery:=le end else begin tlo:=ton;litery:=lon end;
  for j:=yptn to yktn do
    for i:=x-1 to x+2*d do mem[baza:l+2*(i+(j-1)*80)]:=litery+16*tlo;
  end;

```

```

procedure swiatwrsz(x,y,d,kol:byte);
var i,tlo,litery:byte;
begin
  if kol=0 then begin tlo:=te; litery:=le end
  else begin tlo:=ton;litery:=lon end;
  for i:=x-1 to x+d-2 do mem[baza:l+2*(i+(y-1)*80)]:=litery+16*tlo;
end;

```

```

procedure zakresy(zakres:st12;czyzakr,x,yptm1,d:byte);
var i:byte;
begin
  if czyzakr=0 then for i:=1 to length(zakres) do zakres[i]:=hor;
  for i:=x+((d-length(zakres)) div 2)-1 to x-2+((d-length(zakres)) div 2)+length(zakres) do
    mem[baza:2*(i+(yptm1-1)*80)]:=ord(zakres[i-x+((d-length(zakres)) div 2)+1]);
  end;

```

```

procedure czytstringi6(var st:st16;x,y,dt:byte);
var s:string;
  l:byte;
begin
  s:=''; for i:=1 to length(st) do s:=s+st[i];
  czytstring(s,x,y,dt); st:='';
  for i:=1 to length(s) do st:=st+s[i];
end;

```

```

procedure poslug;
begin
  gotoxy(1,25);
  p('cyfra','CYFRA LICZBY '); p(#27+'(be) ','KASOWANIE CYFRY ');
  p('Enter('+#27+#25+#24+#25+' PgUp PgDn Esc)',' KONIEC LICZBY');
  mem[baza:3999]:=lobt+16*tobt; gotoxy(80,24)
end;

```

```

function rdr(wsk : boolean; poprzec,x,y,d : byte; bylo : boolean):string;
var i,j:byte;
  sukces:integer;
  l:string;
  enter:boolean;
  kropka:boolean;
begin
  enter:=false; l:=''; for i:=1 to d do l:=l+' ';
  kolot; gotoxy(x,y); l[1]:=zn; write(l); gotoxy(x+1,y); i:=2;
  if wsk and (poprzec=0) then kropka:=true else kropka:=false;
  while not enter do begin
    gotoxy(x+i-1,y); czytczyn(zn,func);
    if zn=#8 then
      if i>1 then begin
        if not (l[i] in ['0'..'9']) then kropka:=false;

```

```

    i:=i-1; gotoxy(x+i-1,y); write(' ');
    gotoxy(x+i-1,y) end
else dzwiek
else begin
  if (((not func) and (zn in konniefunc))
    or (func and wsk and (zn in znakikonca))) then begin
    enter:=true;
    if (not func) and (zn=#13) then bylo:=false end
  else
    if (not func) and ((zn in poczatekliczby)
      or ((zn='.') and (not kropka))) then begin
      write(zn);
      if zn='.' then kropka:=true;-
      l[i]:=zn; i:=i+1;
      if i=d+1 then begin
        dzwiek; bylo:=false; enter:=true
      end
    end
  end
end
end;
l:=copy(l,i,i-1); rdr:=1
end;

procedure rd(var liczba : real; lpocz : real; poprzec,x,y,d : byte; bylo : boolean);
{ czytanie liczby rzeczywistej;=true - liczba poprawna }
var l:string;
    su:integer;
begin
  l:=rdr(true,poprzec,x,y,d,bylo);
  if length(l)<1 then liczba:=lpocz else val(l,liczba,su);
end;

procedure czytreal(var wart:real;dlicz,poprzec:byte;zakres1,zakres2:real;
  x,y,d:byte;var bylo:boolean);
var liczba : real;
begin {czyt}
  if (((not ((zn in poczatekliczby) or (zn='-')))) and (not func))
    or func) then dzwiek
  else begin
    poslug; bylo:=true; liczba:=wart;
    { jesli byl blad przywraca sie normalne podswietlenie okienka}
    rd(liczba,-1,poprzec,x,y,d,bylo);
    if (liczba<zakres1) or (liczba>zakres2) or (liczba=-1) then begin
      { wpraw. liczba nie miesci sie w zakresie}
      bylo:=false; kursor; swiatlo(x,y,d,2); dzwiek;
      objas(1,'Liczba nie zawiera sie w zadanym zakresie'
        +' - naciśnij dowolny klawisz');
      czytczyn(zn,func); swiatlo(x,y,d,1) end
    else {liczba miesci sie w podstawowym zakresie}
      wart:=liczba; kursor
    end;
end;

procedure ld(var liczba :longint; lpocz :longint; x,y,d : byte; bylo : boolean);
{ czytanie liczby rzeczywistej;=true - liczba poprawna }
var l:string;
    su:integer;
begin
  l:=rdr(false,0,x,y,d,bylo);
  if length(l)<1 then liczba:=lpocz else val(l,liczba,su);
end;

procedure czytword(var wart:word;dlicz,zakres1,zakres2:word;x,y,d:byte;
  var bylo:boolean);
var liczba:longint;

begin {czyt}
  if ((not func) and (not (zn in poczatekliczby))) or func then dzwiek
  else begin
    poslug; bylo:=true; liczba:=wart;
    { jesli byl blad przywraca sie normalne podswietlenie okienka}

```

```

if (liczba<zakres1) or (liczba>zakres2) or (liczba=-1) then begin
  { wpraw. liczba nie mieści się w zakresie}
  bylo:=false; kursor; swiatlo(x,y,d,2); dzwiek;
  objas(1,'Liczba nie zawiera się w zadanym zakresie'
    +' - naciśnij dowolny klawisz');
  czytczyn(zn,func); swiatlo(x,y,d,1) end
else {liczba mieści się w podstawowym zakresie}
  wart:=liczba; kursor
end;
end;

procedure czytbyte(var wart:byte;dlicz,zakres1,zakres2,x,y,d:byte;
  var bylo:boolean);
var liczba:word;
begin
  liczba:=wart; czytword(liczba,dlicz,zakres1,zakres2,x,y,d,bylo); wart:=liczba
end;

procedure czytlongint(var wart:longint;dlicz:byte;
  zakres1,zakres2:longint;x,y,d:byte;var bylo:boolean);
var liczba:longint;
begin {czyt}
  if ((not ((zn in poczatekliczby) or (zn='-')) and (not func)) or func then dzwiek
  else begin
    poslug; bylo:=true; liczba:=wart;
    { jesli byl blad przywraca sie normalne podswietlenie okienka}
    ld(liczba,-2147483647,x,y,d,bylo);
    if (liczba<zakres1) or (liczba>zakres2) or (liczba=-2147483647) then begin
      { wpraw. liczba nie mieści się w zakresie}
      kursor; bylo:=false; swiatlo(x,y,d,2); dzwiek;
      objas(1,'Liczba nie zawiera się w zadanym zakresie'
        +' - naciśnij dowolny klawisz');
      czytczyn(zn,func); swiatlo(x,y,d,1) end
    else {liczba mieści się w podstawowym zakresie}
      wart:=liczba; kursor
    end;
  end;

procedure czytinteger(var wart:integer;dlicz:byte;
  zakres1,zakres2:integer;x,y,d:byte;var bylo:boolean);
var liczba:longint;
begin
  liczba:=wart; czytlongint(liczba,dlicz,zakres1,zakres2,x,y,d,bylo); wart:=liczba
end;

procedure czytshortint(var wart:shortint;dlicz:byte;
  zakres1,zakres2:shortint;x,y,d:byte;var bylo:boolean);
var liczba:longint;
begin
  liczba:=wart; czytlongint(liczba,dlicz,zakres1,zakres2,x,y,d,bylo); wart:=liczba
end;

procedure czytdyskretnb(var wart:byte;od,doo:byte);
begin
  if zn<>' ' then dzwiek
  else if wart=doo then wart:=od else wart:=wart+1
end;

procedure czytdyskretni(var wart:integer;od,doo:integer);
begin
  if zn<>' ' then dzwiek else
  if wart=doo then wart:=od else wart:=wart+1
end;

procedure przewin(x1,y1,x2,y2,kier:byte);
var x,y:integer;
begin
  if y1<=y2 then
    for y:=y2#integer(kier<0)+y1#integer(kier>0)-1 downto (y2#integer(kier>0)+y1#integer(kier<0))-1 do

```

end;

```

procedure ifdluzszy(txt:string; dlug :byte);
begin
  if length(txt)>dlug then dlug:=length(txt);
end;

```

```

function oknopytania(illin:byte;txtg,txt1,txt2,txt3,txt4,txt5,txt6,txt7:string;
                    propozycja:a4st;ilosc:byte):byte;

```

```

var x1,y1,x2,y2,dlug,i,dlwart : byte;
    pomekr : ekran;
    enter : boolean;
    pocz : array [1..4] of byte;

```

begin

pamietajekr(pomekr);

y1:=13-((illin+4) div 2); y2:=y1+illin+3; dlug:=length(txtg);

ifdluzszy(txt1,dlug); ifdluzszy(txt2,dlug); ifdluzszy(txt3,dlug); ifdluzszy(txt4,dlug);

ifdluzszy(txt5,dlug); ifdluzszy(txt6,dlug); ifdluzszy(txt7,dlug);

if ilosc<>0 then begin dlwart:=ilosc-1;

for i:=1 to ilosc do dlwart:=dlwart+length(propozycja[i])+2;

end;

if dlwart>dlug then dlug:=dlwart;

dlug:=dlug+12; x1:=41-(dlug div 2); x2:=x1+dlug-1;

kolw; bezobw(x1,y1,x2,y2); oknoobw(x1+3,y1+1,x2-3,y2-1);

pisztxt(y1+1,txtg); pistxt(y1+2,txt1); pistxt(y1+3,txt2); pistxt(y1+4,txt3);

pisztxt(y1+5,txt4); pistxt(y1+6,txt5); pistxt(y1+7,txt6); pistxt(y1+8,txt7);

if ilosc<>0 then begin pocz[1]:=41-(dlwart div 2);

for i:=2 to ilosc do pocz[i]:=pocz[i-1]+length(propozycja[i-1])+3;

swiec(pocz[1],propozycja[1],1);

for i:=2 to ilosc do swiec(pocz[i],propozycja[i],0);

i:=1; enter:=false; objas(1,'Wybierz odpowiedz');

while not enter do begin

czytczyn(zn,func);

if func then

case zn of

#75:begin

swiec(pocz[i],propozycja[i],0); if i=1 then i:=ilosc else i:=i-1; swiec(pocz[i],propozycja[i],1)

end;

#77:begin

swiec(pocz[i],propozycja[i],0); if i=ilosc then i:=1 else i:=i+1; swiec(pocz[i],propozycja[i],1)

end;

end

else

case zn of

#13:enter:=true;

#27:begin

enter:=true; i:=0;

end

end

end; oknopytania:=i;

end; kursor; odtworzekr(pomekr)

end;

```

procedure oknosrodk(illin:byte;txtg,txt1,txt2,txt3,txt4,txt5,txt6,txt7:string);

```

var x1,y1,x2,y2,dlug,i:byte;

enter:boolean;

begin

y1:=13-((illin+4) div 2); y2:=y1+illin+3; dlug:=length(txtg);

ifdluzszy(txt1,dlug); ifdluzszy(txt2,dlug); ifdluzszy(txt3,dlug); ifdluzszy(txt4,dlug);

ifdluzszy(txt5,dlug); ifdluzszy(txt6,dlug); ifdluzszy(txt7,dlug);

dlug:=dlug+12; x1:=41-(dlug div 2); x2:=x1+dlug-1;

kolw; bezobw(x1,y1,x2,y2); oknoobw(x1+3,y1+1,x2-3,y2-1);

pisztxt(y1+1,txtg); pistxt(y1+2,txt1); pistxt(y1+3,txt2); pistxt(y1+4,txt3);

pisztxt(y1+5,txt4); pistxt(y1+6,txt5); pistxt(y1+7,txt6); pistxt(y1+8,txt7);

kursor;

end;

```

procedure piszspacje(y:byte);

```

var i:byte;

```
for i:=1 to 78 do mem[baza:(i+(y-1)*80)*2]:=32  
end;
```

```
function raportnaekranie:byte;  
const a:a4st=(' TAK ', ' NIE ', '', '');  
begin  
  raportnaekranie:=oknopytania(5, '', '', 'Czy chcesz przegladac', 'na ekranie tabele raportow?', '', '', '', 'a,2)  
end;
```

```
end.
```

~~11/11~~

