AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

Do użytku wewnętrznego

Egz. pojed.

1

SYSTEM MODELOWANIA WALKI ZBROJNEJ

"MODEL - 1"

ANIMACJA UGRUPOWAŃ

część III

WARSZAWA 1989

AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

Do użytku wewnętrznego

Egz.pojed.

1

SYSTEM MODELOWANIA WALKI ZBROJNEJ

"MODEL - 1"

ANIMACJA UGRUPOWAŃ

część III

WARSZAWA 1989

AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAN STRATEGICZNO-OBRONNYCH

Do użytku wewnętrznego

Egz. pojedynczy

SYSTEM MODELOWANIA WALKI ZBROJNEJ

" MODEL - 1 "

ANIMACJA UGRUPOWAŃ

część III

1

5/1087

AUTORZY:

płk R.KULCZYCKI - zadania projektowe oraz projekt koncepcyjny

ppłk T.STASZCZYK, ppłk S.DĄBROWSKI, mjr M.PSZCZÓŁKA - algorytmizacja i oprogramowanie.

Warszawa 1989

Spis treści:

1. Metoda obliczania współrzędnych str. 3
2. Algorytm animacji str. 9
3. Tabulogram programu animacji str. 27
4. Przykładowe wydruki problemu animacji str. 37

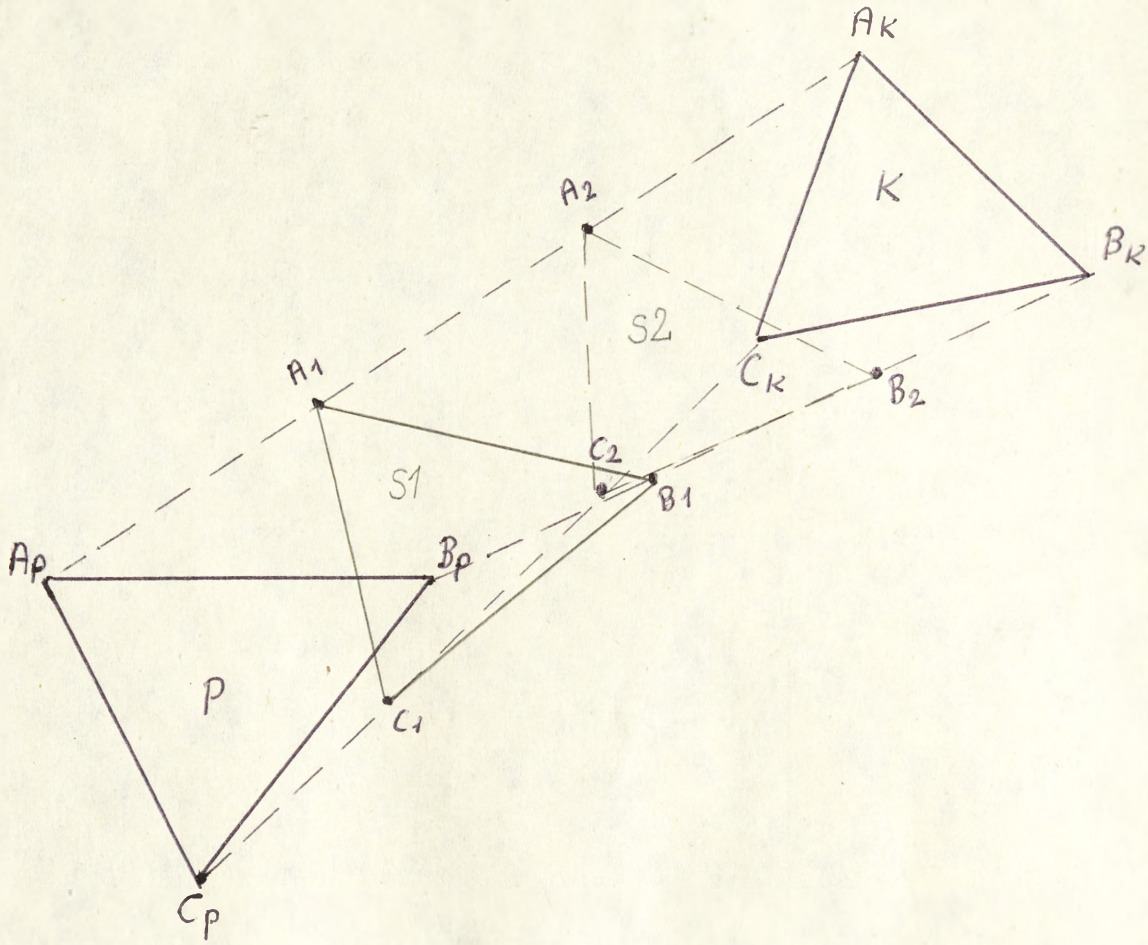
METODA
OBLICZANIA WSPÓŁRZĘDNYCH.

METODYKA OKREŚLANIA PARAMETRÓW ANIMACJI

Przy określaniu parametrów animacji muszą być spełnione następujące warunki:

- muszą być podane położenia pośrednie pomiędzy położeniami zasadniczymi (początkowym i końcowym);
- maksymalna ilość położeń pośrednich nie może być większa od 10;
- minimalna ilość położeń pośrednich nie może być mniejsza od 2;
- prędkość poruszania się poszczególnych ugrupowań pomiędzy położeniami pośrednimi musi być stała, natomiast na poszczególnych etapach animacji może się zmieniać;
- prędkości poruszania się poszczególnych ugrupowań mogą być rożne;
- możliwe jest zanikanie poszczególnych ugrupowań, powstawanie nowych ugrupowań i ich elementów;
- zmiana poszczególnych punktów opisujących ugrupowanie jest liniowa na całym odcinku pomiędzy położeniami pośrednimi natomiast nie musi być liniowa na całym etapie działań;
- przyjęty typ ugrupowania musi być stały w całym zakresie prowadzonej animacji.

Współrzędne poszczególnych punktów opisujących ugrupowanie obliczane są na podstawie zmian liniowych. Metodę obliczania tych współrzędnych przedstawiono na poniższym rysunku.



Rys. 1.

Określanie pośrednich położeń w czasie animacji.

Na rysunku oznaczono:

P - położenie początkowe;

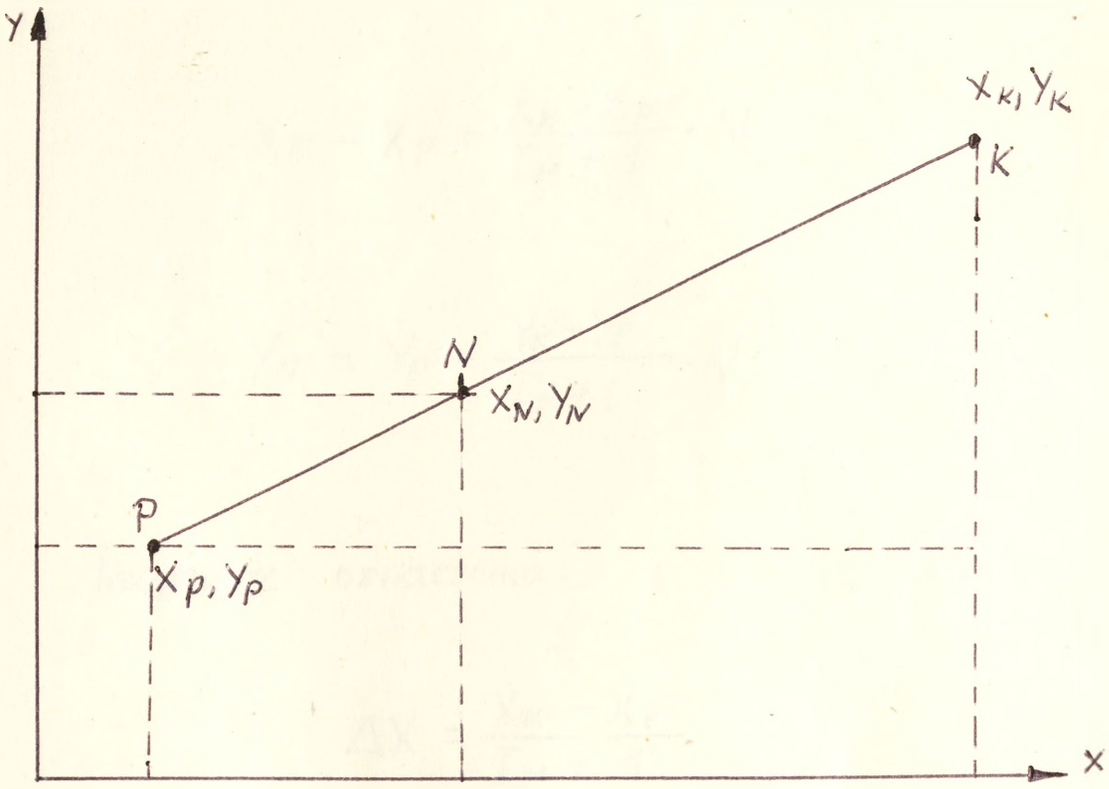
K - położenie końcowe;

S - położenie pośrednie

Kolejne położenia pośrednie S1, S2 są tak wyznaczane, że punkty opisujące daną figurę geometryczną odpowiadające sobie leżą zawsze na linii prostej łączącej te punkty w położeniu początkowym i końcowym.

Przyjmując liniową zmianę położenia oraz znajomość położenia początkowego i końcowego możemy obliczyć współrzędne położeń pośrednic.

Sytuację taką przedstawiono na poniższym rysunku.



Rys 2.

Położenie punktów pośrednich przy animacji.

Wielkości dane:

1) położenie początkowe:

X_P, Y_P

2) położenie końcowe:

X_K, Y_K

3) ilość położeń pośrednich:

I_M

Wielkości szukane:

współrzędne dowolnego N-tego położenia pośredniego:

$$X_N, Y_N$$

$$X_N = X_P + \frac{X_K - X_P}{I_M + 1} \cdot N$$

$$Y_N = Y_P + \frac{Y_K - Y_P}{I_M + 1} \cdot N$$

Przyjmując oznaczenia:

$$\Delta X = \frac{X_K - X_P}{I_M + 1}$$

$$\Delta Y = \frac{Y_K - Y_P}{I_M + 1}$$

Wyrażenia na współrzędne możemy napisać w postaci:

$$X_N = X_P + \Delta X \cdot N$$

$$Y_N = Y_P + \Delta Y \cdot N$$

Najczęściej w praktycznych zastosowaniach używa się tych wzorów w postaci rekurencyjnej:

$$X_{N+1} = X_N + \Delta X$$

$$Y_{N+1} = Y_N + \Delta Y .$$

ALGORYTM ANIMACJI

ALGORYTM ANIMACJI

START

↓

Wyświetl na ekranie komputera wykaz funkcji realizowanych przez system.

↓

Wybierz odpowiednią funkcję przez wprowadzenie do komputera odpowiadającej jej liczbie:

1 - opis komend pracy na mapie;
2 - praca na mapie;
3 - tworzenie ugrupowań;
4 - korekta ugrupowań;
5 - kopiowanie sytuacji w komputerze;
6 - kasowanie ugrupowań;
7 - przeglądanie;
8 - drukowanie;
9 - odczyt sytuacji bojowej z dyskietki;

↓

(1)

(1)

10 - zapis sytuacji bojowej na dyskietke;
11 - parametry animacji;
0 - koniec pracy;

Wprowadź do komputera liczbę 11.

Wprowadź do komputera parametry animacji:

Czas trwania jednej sytuacji w godz.
Czas trwania jednej sytuacji jest to czas odpowiadający w rzeczywistości czasowi w jakim dane ugrupowanie przejdzie z jednego położenia w drugie.

(2)

(2)

Ilość obrazów animacji w jednej sytuacji.

Parametr ten określa ile położeń pośrednich ma być wyświetlonych na ekranie przy przejściu danego ugrupowania z jednego położenia w drugie.

Czas trzymania obrazu animacji w msek.

Czas trzymania obrazu jest to czas podany w milisekundach, wyświetlenia na ekranie monitora jednego położenia pośredniego, zadanych ugrupowań.

W algorytmie przyjęto oznaczenia.

T_{SP} - czas trwania jednej sytuacji w godz.

(3)

(3)

↓

Nsp - ilość obrazów animacji w jednej sytuacji.

Two - czas trwania obrazu w milisekundach.

↓

Wprowadź do komputera dynamiczne parametry animacji:

Pps - początkowy numer sytuacji bojowej od której należy rozpocząć animację.

Ls - ilość sytuacji która ma być uwzględniona przy animacji.

↓

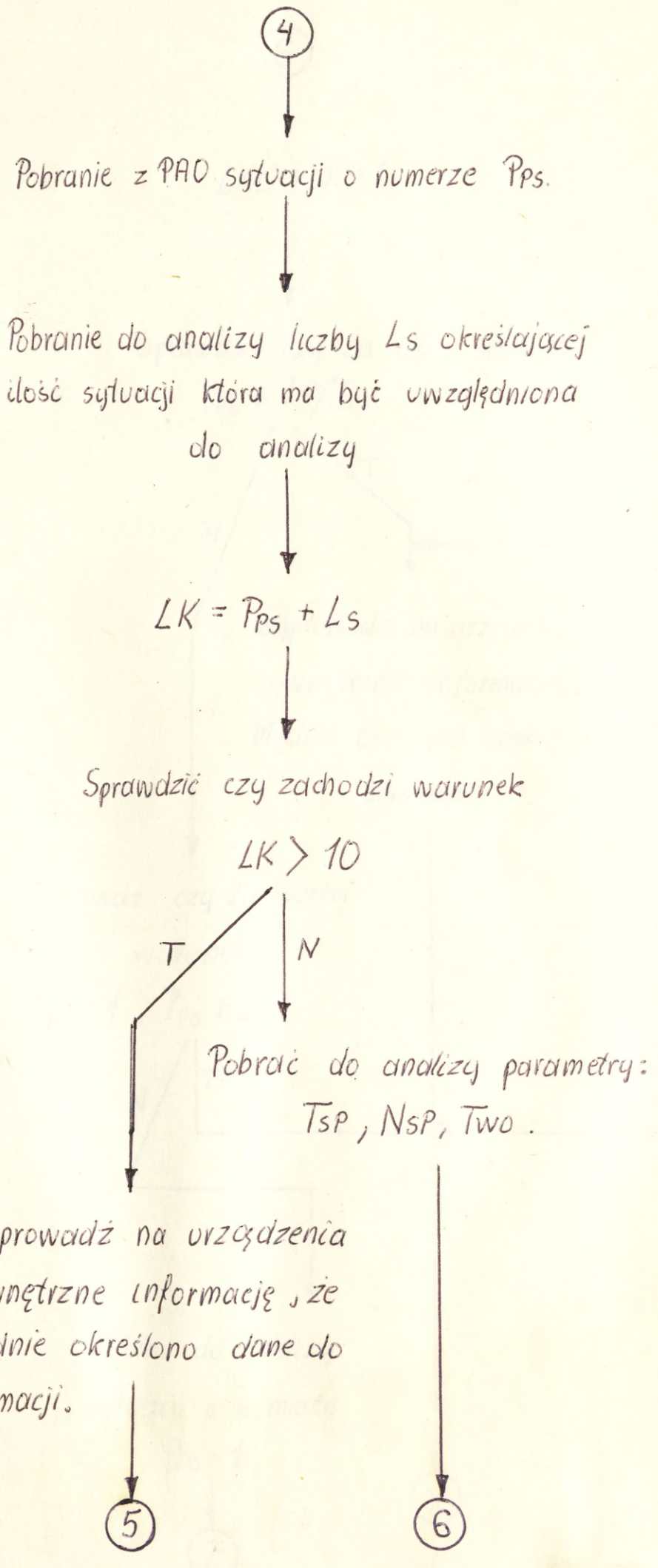
Wprowadzenie do komputera sygnału rozpoczęcia animacji.

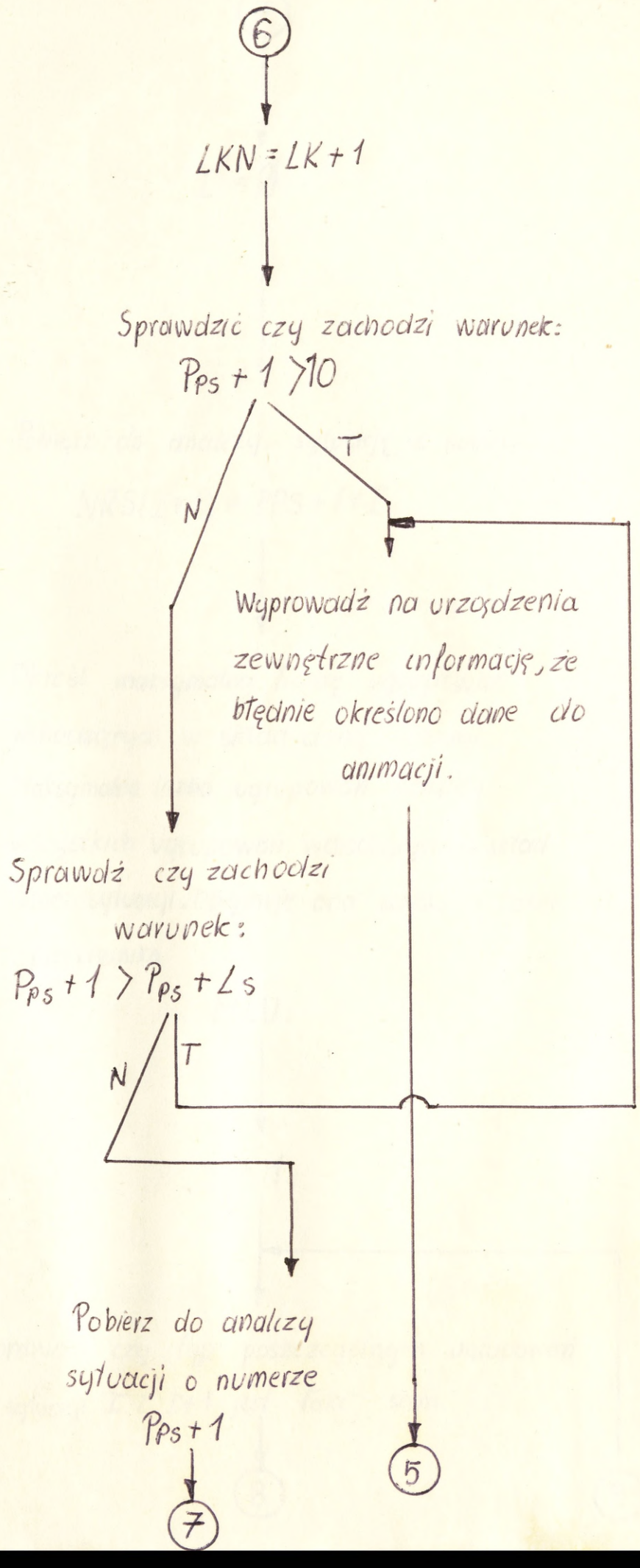
↓

Sprawdzenie warunków realizacji animacji.

↓

(4)


4
Pobranie z PAO sytuacji o numerze Pps.
Pobranie do analizy liczby Ls określającej ilość sytuacji która ma być uwzględniona do analizy
LK = Pps + Ls
Sprawdzić czy zachodzi warunek
LK > 10
T
N
Pobrać do analizy parametry: TsP, NsP, Two.
Wyprowadź na urządzenia zewnętrzne informację, że błędnie określono dane do animacji.
5
6


6
LKN = LK + 1
Sprawdzić czy zachodzi warunek:
Pps + 1 > 10
T
N
Wyprowadź na urządzenia zewnętrzne informację, że błędnie określono dane do animacji.
Sprawdź czy zachodzi warunek:
Pps + 1 > Pps + Ls
N
T
Pobierz do analizy sytuacji o numerze
Pps + 1
5
7

(7)

↓

I = Ø

↓

Pobierz do analizy sytuację o numerze

NRS(I+1) = PPS + 1 + I

↓

Określ maksymalną liczbę ugrupowań wchodzących w skład danej sytuacji. Maksymalna liczba ugrupowań dotyczy wszystkich ugrupowań wchodzących w skład danej sytuacji. Obejmuje ona wojska własne i przeciwnika.

MLU.

↓

J = 1

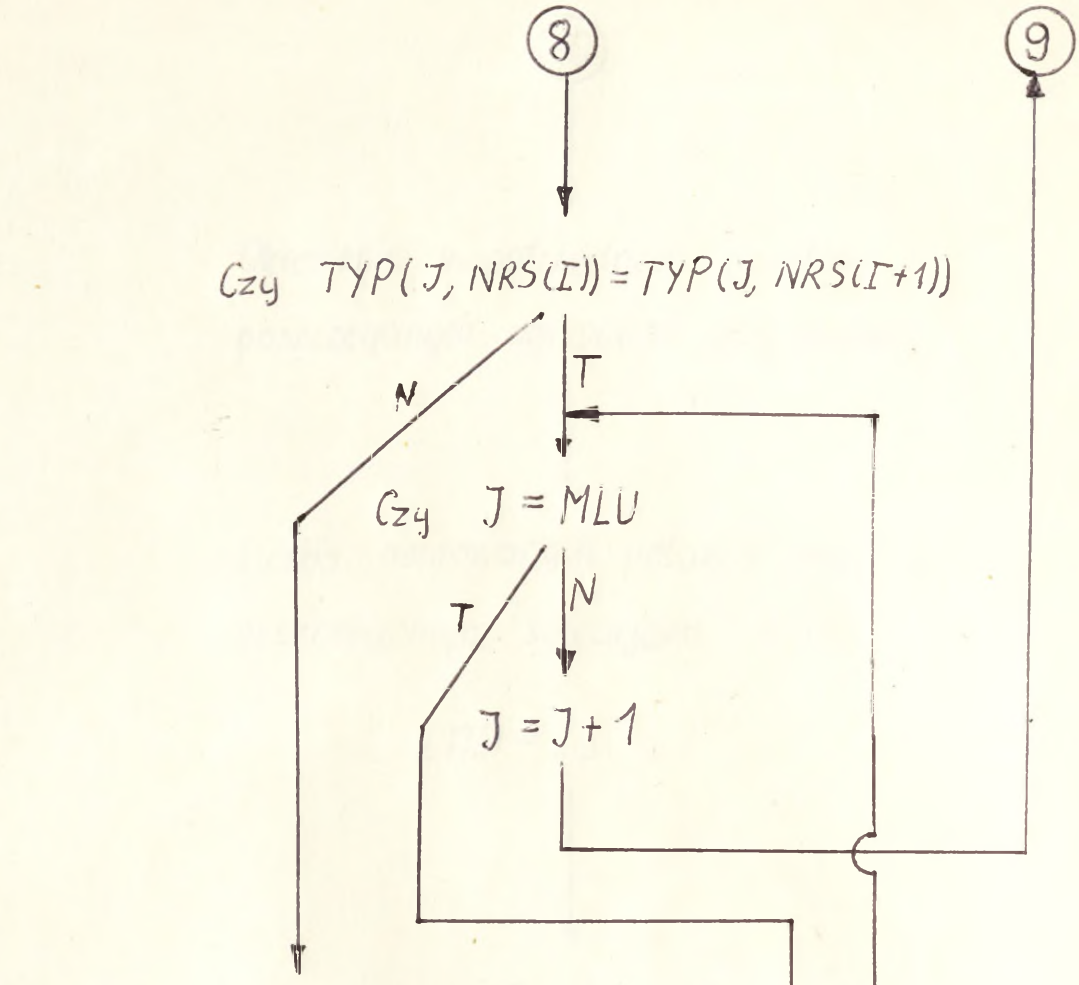
↓

Sprawdź czy typ poszczególnych ugrupowań w sytuacji I i I+1 jest taki sam.

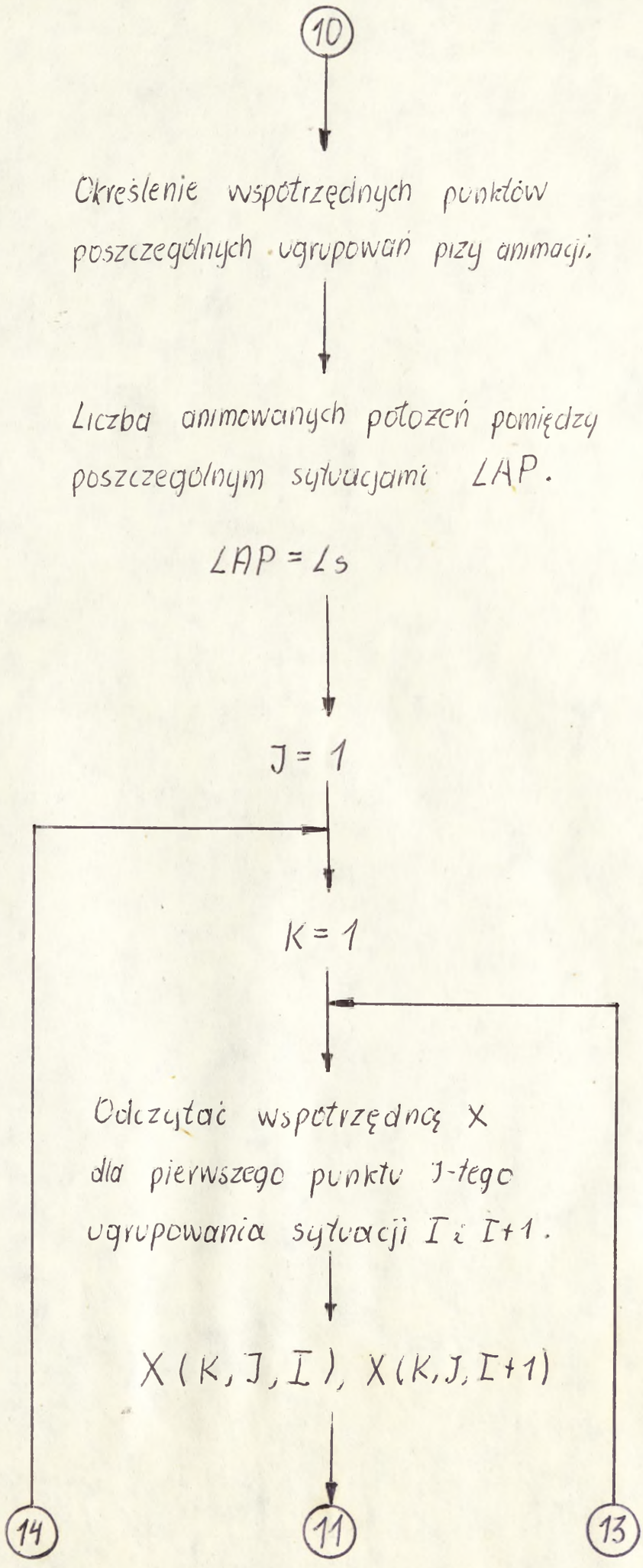
↓

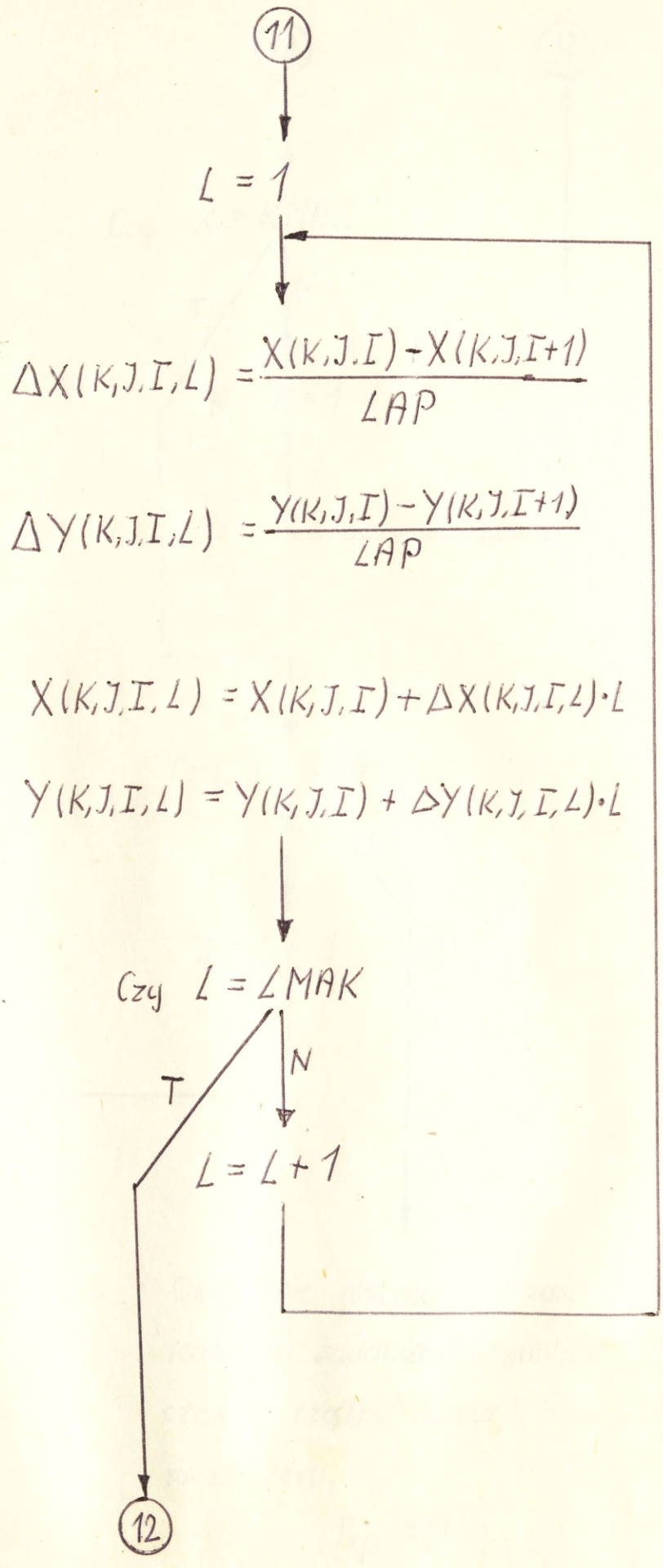
(8)

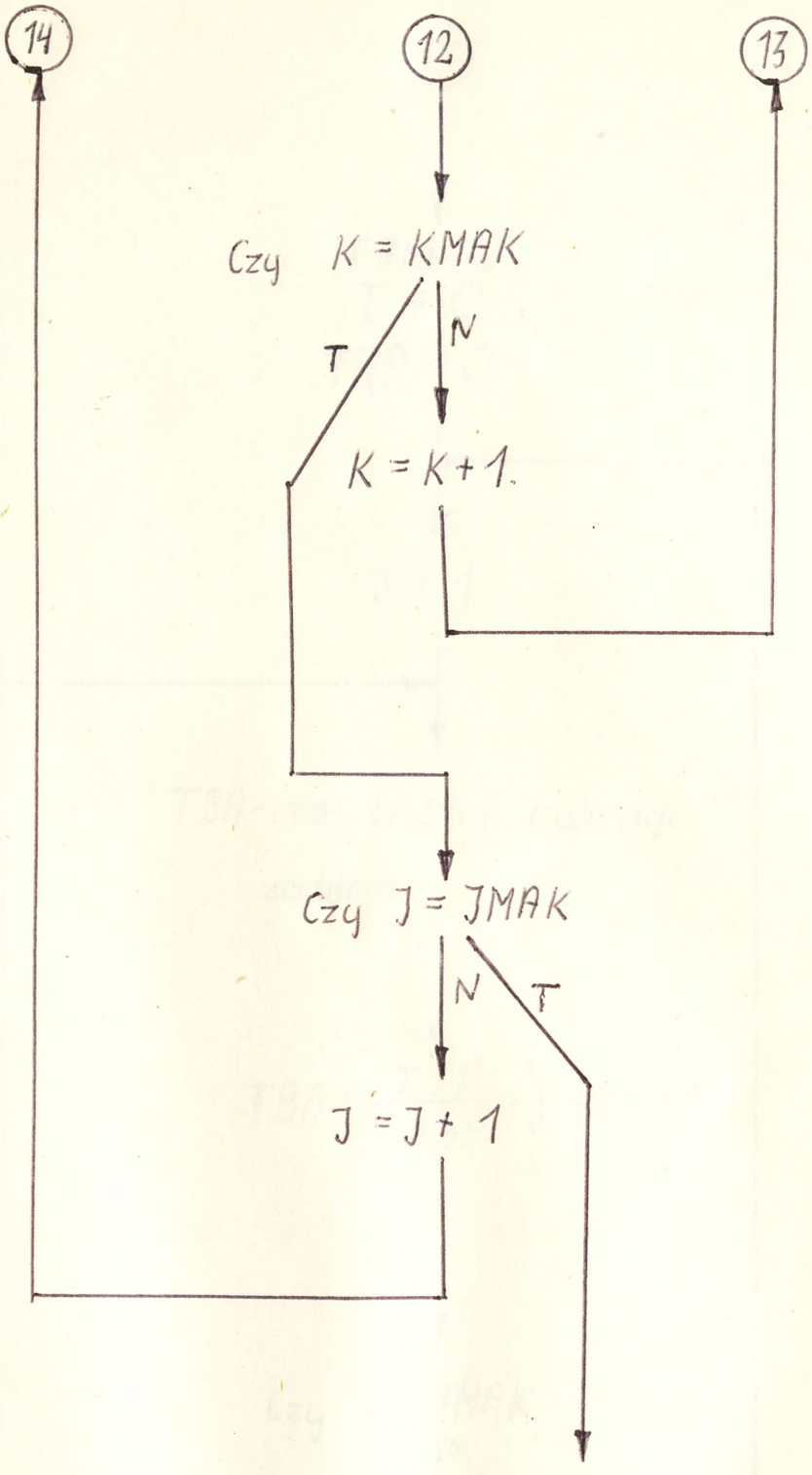
(9)



Przyjmij współrzędne poszczególnych punktów opisujących ugrupowanie że dla położenia I+1 są takie same jak współrzędne środka ugrupowania położenia I oraz współrzędne punktów położenia I dla następnego typu ugrupowania, są takie same jak współrzędne środka tego ugrupowania położenia I+1. Prowadzi to do zaniku ugrupowania w położeniu I i powstanie nowego ugrupowania w położeniu I+1.

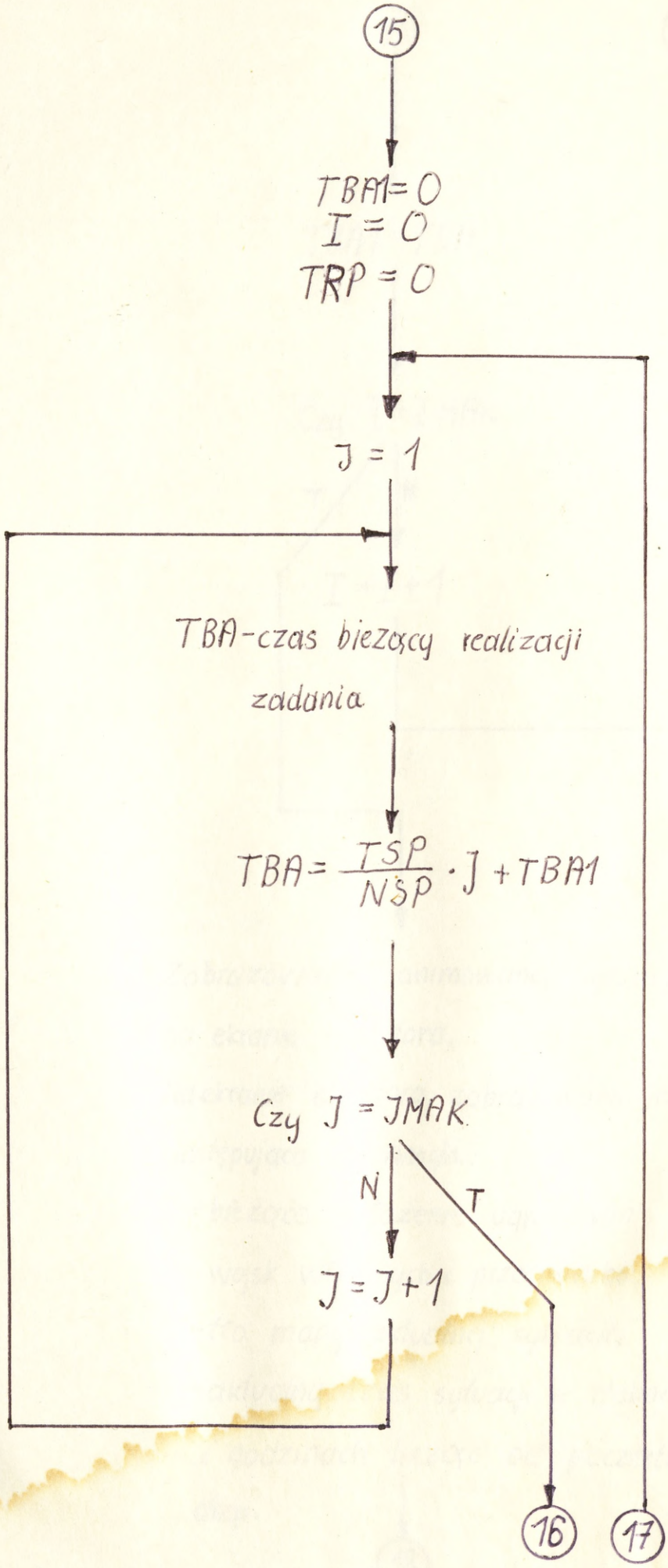

10
Określenie współrzędnych punktów poszczególnych ugrupowań przy animacji.
Liczba animowanych położeń pomiędzy poszczególnym sytuacjami LAP.
LAP = Ls
J = 1
K = 1
Odczytać współrzędną X dla pierwszego punktu J-tego ugrupowania sytuacji I i I+1.
X(K,J,I), X(K,J,I+1)
14
11
13

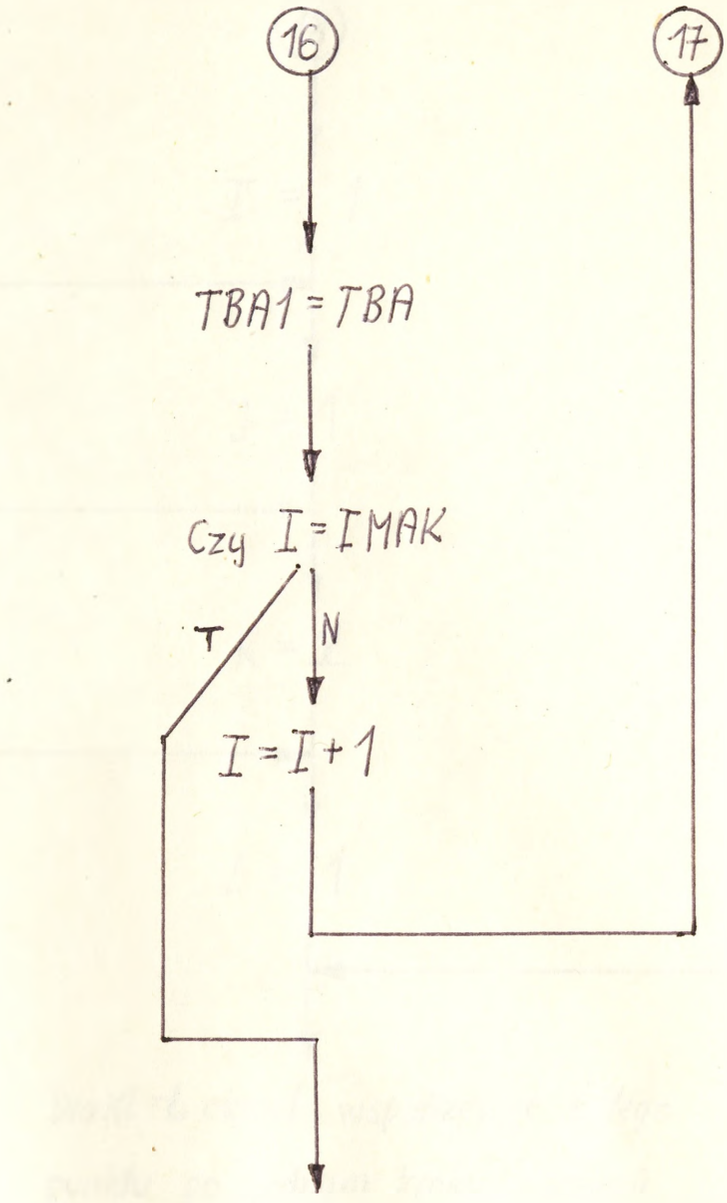

11
L = 1
ΔX(K,J,I,L) = (X(K,J,I) - X(K,J,I+1)) / LAP
ΔY(K,J,I,L) = (Y(K,J,I) - Y(K,J,I+1)) / LAP
X(K,J,I,L) = X(K,J,I) + ΔX(K,J,I,L)·L
Y(K,J,I,L) = Y(K,J,I) + ΔY(K,J,I,L)·L
Czy L = LMAK
T
N
L = L + 1
12



Określenie aktualnego czasu realizacji zadania, przyjmując czas początku operacji Trp równy zeru.

$Trp = 0$

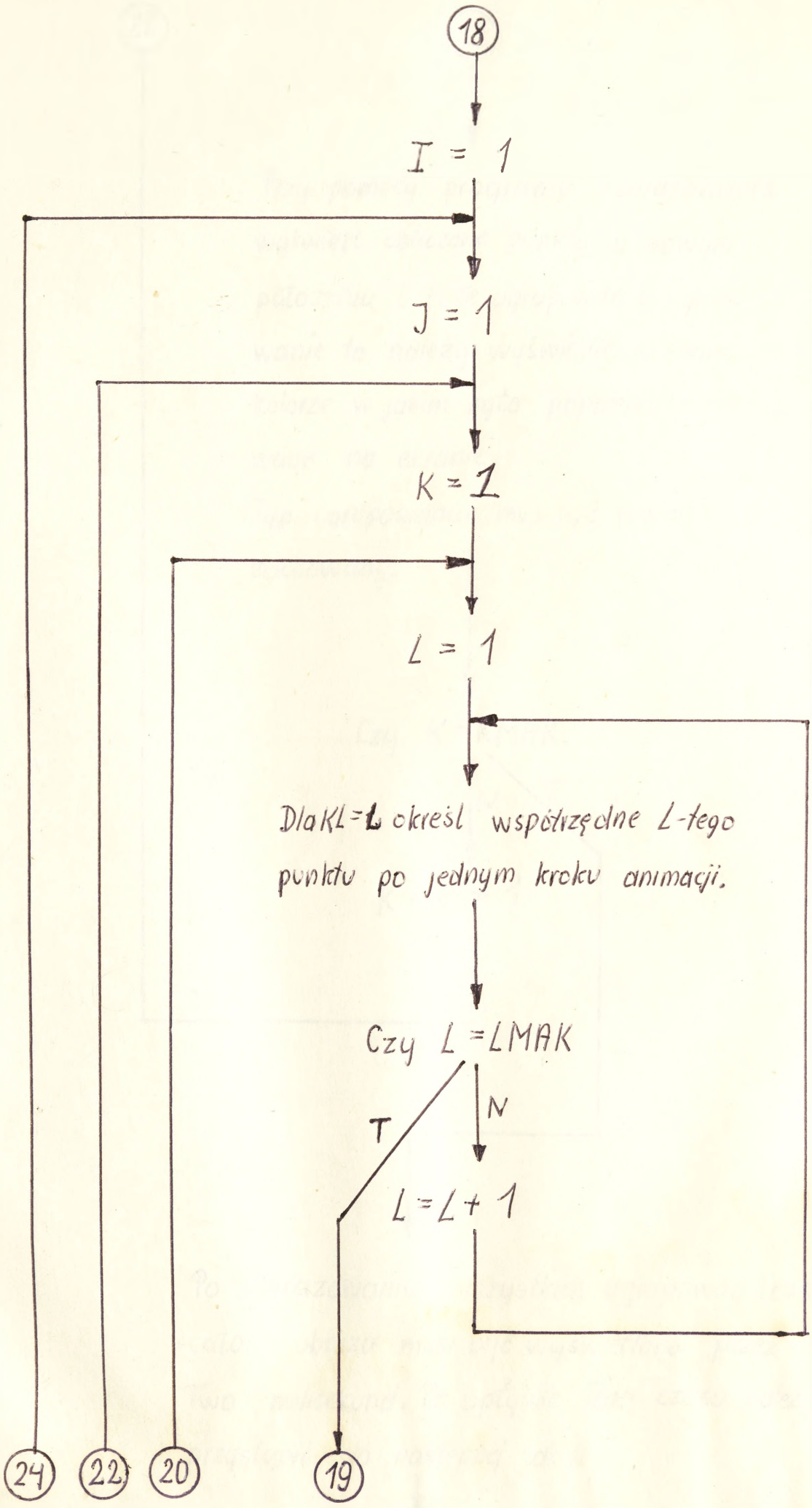

15
TBA1=0
I=0
TRP=0
J=1
TBA-czas biezacy realizacji zadania
TBA=TSP/NSP·J+TBA1
Czy J=JMAK
N
T
J=J+1
16
17



Zobrazowanie animowanej sytuacji na ekranie monitora.

Na ekranie monitora zobrazowana jest następująca informacja:

- bieżące położenie ugrupowań wojsk własnych i przeciwnika,
- tło mapy aktualnej sytuacji,
- aktualny czas sytuacji w dobach i godzinach licząc od początku akcji.


18
I = 1
J = 1
K = 1
L = 1
Dla KL=L określ współrzędne L-tego punktu po jednym kroku animacji.
Czy L = LMAK
T
N
L = L + 1
24
22
20
19

(20)

(19)

Przy pomocy programu zobrazowania wyświetl obliczone punkty w nowym położeniu i całe ugrupowanie. Ugrupowanie to należy wyświetlić w takim kolorze w jakim było poprzednio zobrazowane na ekranie.

Typ ugrupowania musi być również zachowany.

Czy K = KMAK.

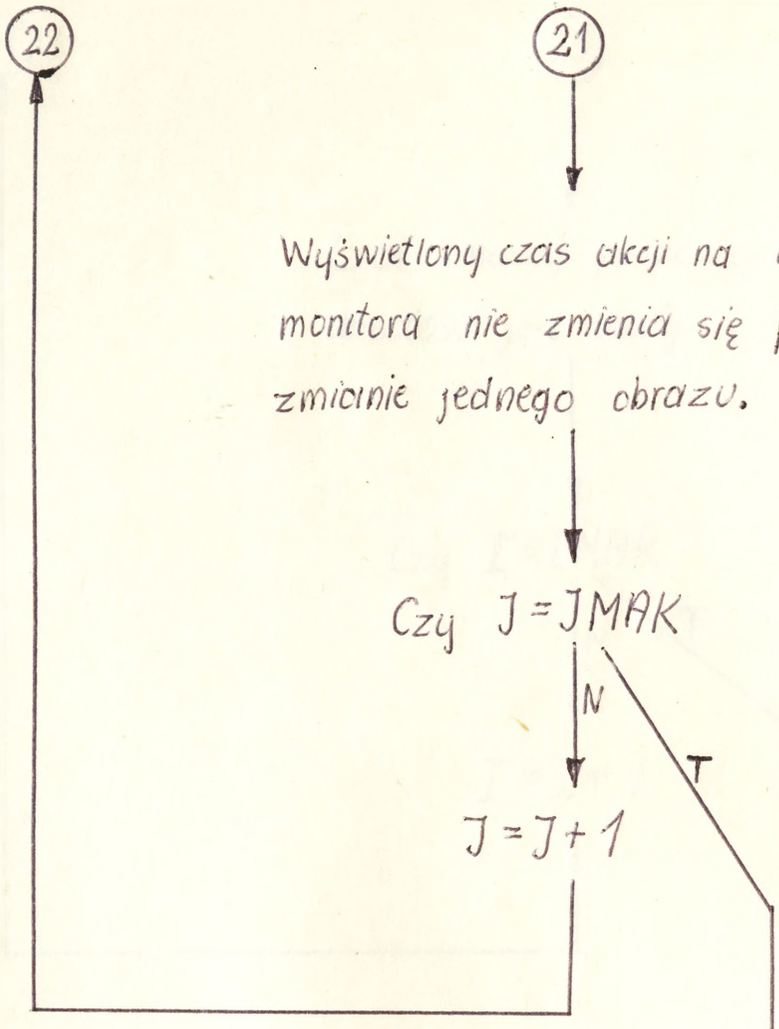
N

T

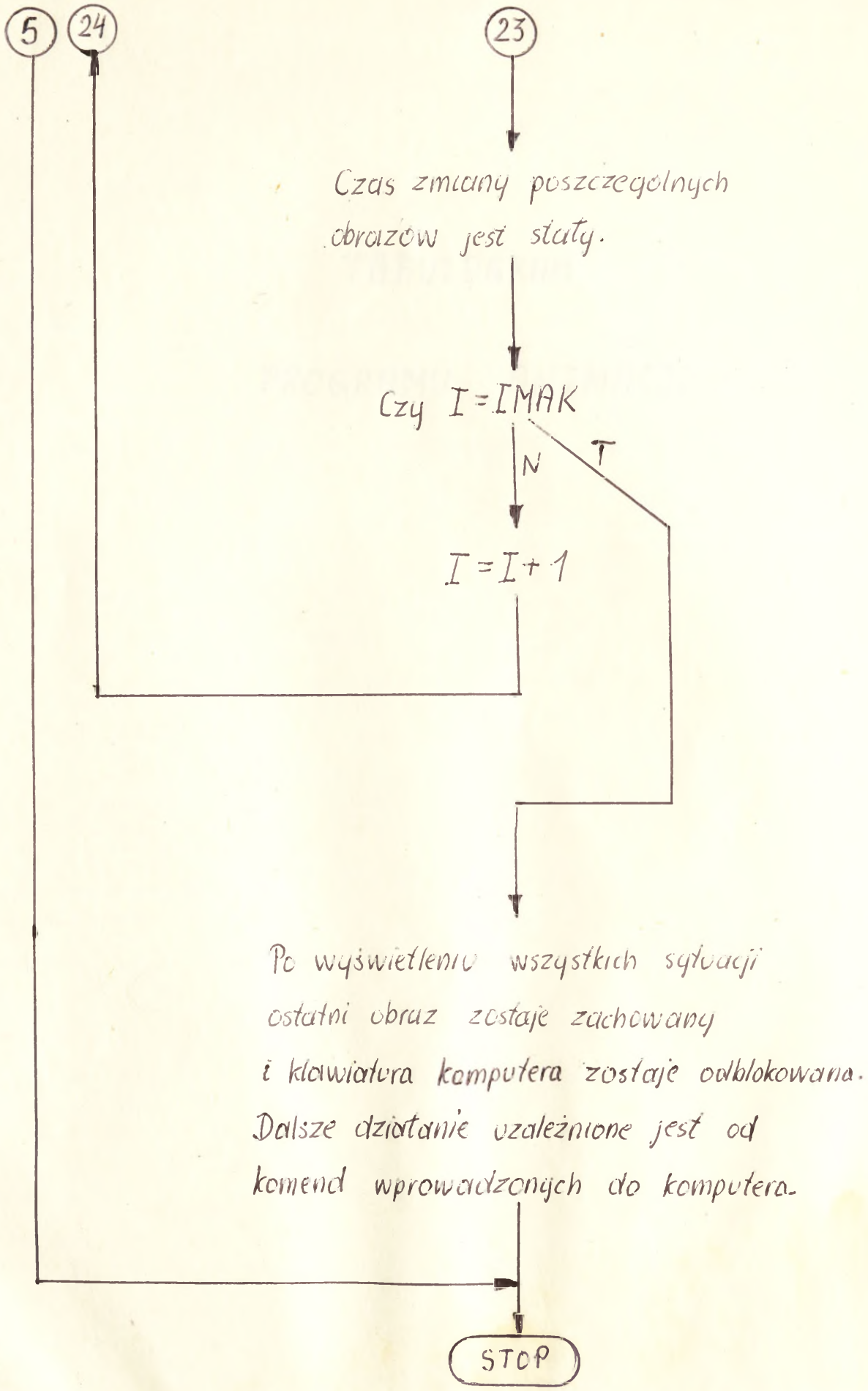
K = K + 1

Po zobrazowaniu wszystkich ugrupowań i czasu, całość obrazu musi być wyświetlana przez Two milisekund. Po upływie Tego czasu należy przystąpić do następnej akcji.

(21)



Po wyświetleniu pełnej sytuacji należy wygasić sytuację poprzednią i przystąpić do realizacji następnych. W czasie wyświetlania wszystkich sytuacji parametry ruchu są stałe i nie ulegają zmianie.


5
24
23
Czas zmiany poszczególnych obrazów jest stały.
Czy I=IMAK
N
T
I=I+1
Po wyświetleniu wszystkich sytuacji ostatni obraz zostaje zachowany i klawiatura komputera zostaje odblokowana. Dalsze działanie uzależnione jest od komend wprowadzonych do komputera.
STOP

TABULOGRAM

PROGRAMU ANIMACJI

```
begin
 znak:= #$7e;
 for krok:=1 to mkrok do
 swietl[krok]:=1;
 animstep:=12;
 lad:=1;
 powietrze:=1;
 woda:=1;
 numermapy:=0;
 wskmapy:=3;
 for krok:=1 to mkroki do
 begin
  new(walka[krok]);
  fillchar(walka[krok]^.w[1,1,1],2*mstrona*mgrupa*mblok,0);
 end;

 repeat
  clrscr;
  writeln('Wybierz funkcje :');
  writeln('  1 -opis komend pracy na mapie');
  writeln('  2 -praca na mapie');
  writeln('  3 -tworzenie ugrupowan');
  writeln('  4 -korekta ugrupowan');
  writeln('  5 -kopiowanie sytuacji w komputerze');
  writeln('  6 -kasowanie ugrupowan');
  writeln('  7 -przegladanie');
  writeln('  8 -drukowanie');
  writeln('  9 -odczyt sytuacji bojowej z dyskietki');
  writeln(' 10 -zapis sytuacji bojowej na dyskietke');
  writeln(' 11 -parametry animacji');
  writeln('  0 -koniec pracy');
  write('Wpisz numer funkcji : ');
  {$i-}readln(wybor){$i+};
  if ioresult=0 then
  case wybor of
   1:komendy;
   2:mapa;
   3:tworzenie;
   4:korekta;
   5:kopiuj;
   6:kasowanie;
   7:przegladanie;
   8:wydruk;
   9:pobierz;
   10:schowaj;
   11:paranim;
  end;
 until wybor=0;
 for krok:=1 to mkroki do release(walka[krok]);
 clrscr;
end.
```

```
procedure komendy;
begin
 clrscr;
 writeln('OPIS KOMEND SYTUACJI BOJOWEJ');
 writeln(' Komendy obslugi mapy:');
 writeln('i = [0..9] = rysowanie mapy numer i');
 writeln('  Stan i=',numermapy);
 writeln('M = wyswietlenie lub wygaszenie mapy');
 write('  Stan: M=');
 if wskmapy=0 then writeln('wygaszona') else writeln('wyswietlana');
 writeln('strzalka[gora,dol,lewo,prawo] = obcinanie mapy');
 writeln('spacja = wyswietlenie powiekszenia obcietej mapy');
 writeln(' Komendy obslugi sytuacji bojowej:');
 writeln('Si = [S1..S9,S0] = wyswietlenie lub wygaszenie sytuacji bojowej nu
 writeln('  Stan wyswietlonych:');
 for krok:=1 to mkrok do
 if swietl[krok]=2 then write(krok,',');
 writeln('');
 writeln('L,P,W = wyswietlenie lub wygaszenie ladowej,powietrznej lub morski
 writeln('       sytuacji bojowej tj. numerow ugrupowan:1..49,50..79,80..99'
 write('  Stan: L=');
 if lad=0 then writeln('wygaszona') else writeln('wyswietlana');
 write('  Stan: P=');
 if powietrze=0 then writeln('wygaszona') else writeln('wyswietlana');
 write('  Stan: W=');
 if woda=0 then writeln('wygaszona') else writeln('wyswietlana');
 writeln('Ai = [A1..A9] = wykonanie i animacji zaczynajac od najwyzszej');
 writeln('                wyswietlonej sytuacji');
 write('  Najwyzsza wyswietlona sytuacja:');
 i:=0;
 for krok:=1 to mkrok do if swietl[krok]=2 then i:=krok;
 writeln(i);writeln('');
 writeln('Wcisnij dowolny klawisz');
 repeat until keypressed;
end;

procedure paranim;
begin
repeat
 clrscr;
 writeln('PARAMETRY ANIMACJI');
 write('Czas trwania jednej sytuacji:',godzkrok,'godz.,nowa wartosc:');
 {$i-}readln(godzkrok){$i+};
 if ioresult=0 then
 begin
  write('Ilosc obrazow animacji w jednej sytuacji:',animstep,'nowa wartosc:'
  {$i-}readln(animstep){$i+};
  if ioresult=0 then
  begin
   write('Czas trzymania obrazu animacji:',break,'[msek],nowa wartosc:');
   {$i-}readln(break){$i+};
  end;
 end;
until ioresult=0;
end;

{*** sterowanie ***}
var
wybor:integer;
```

```
   for strona:=1 to mstrona do
   begin
    if lad=1 then for grupa:=minl to maxl do dajanim;
    if powietrze=1 then for grupa:=minp to maxp do dajanim;
    if woda=1 then for grupa:=minw to maxw do dajanim;
   end;
   krok:=krokanim;
   for astep:=1 to animstep do
   begin
   for strona:=1 to mstrona do
   begin
    gotoxy(36,1);
    czas:=(krokanim-1)*godzkrok+((astep-1)*godzkrok)/animstep;
    dzien:=trunc(czas/24);
    godz:=trunc(czas-dzien*24);
    write(dzien:2,':',godz:2);
    if lad=1 then for grupa:=minl to maxl do robanim;
    if powietrze=1 then for grupa:=minp to maxp do robanim;
    if woda=1 then for grupa:=minw to maxw do robanim;
   end;
   delay(break);
   end;
   krok:=krokanim;
   kolor:=0;
   oznakuj;
   swietl[krokanim]:=1;
   krokanim:=krokanim+1;
   krok:=krokanim;
   kolor:=3;
   oznakuj;
   swietl[krokanim]:=2;
  end;
 end;
end;

{**** mapa ****}

procedure mapa;
begin
 znak:= #$7e;
 mapanr(numermapy);
 while znak<>#$0d do    {cr}
 begin
  read(kbd,znak);
  case znak of
  'a':  {animacja}
  begin
   read(kbd,znak);
   if ((znak>=#$31)and(znak<=#$39)) then
   begin
    move(znak,animile,1);
    animile:=animile-$30;
    animacja;
   end;
  end;
 end;
end;
textmode;
end;
```

```
end;

procedure robanim;
var ksz1,ksz2:integer;
begin
{wymazanie starego obrazka}
 ksz1:=walka[mkrok+1]^.w[strona,grupa,ksztalt];
 ksz2:=walka[mkrok+2]^.w[strona,grupa,ksztalt];
 if astep=1 then
 begin
  kolor:=0;
  krok:=krokanim;
  obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
 end
 else
 begin
 if ksz1<>0 then
 begin
  kolor:=0;
  krok:=mkrok+1;
  obrazek(ksz1);
 end;
 if ksz2<>0 then
 begin
  kolor:=0;
  krok:=mkrok+2;
  obrazek(ksz2);
 end;
 end;
 if astep<>animstep then
 begin
  if ksz1<>0 then zmien(mkrok+1,krokanim+1);
  if ksz2<>0 then zmien(mkrok+2,mkrok+3);
 end
 else
 begin
  if walka[krokanim+1]^.w[strona,grupa,ksztalt]<>0 then
  begin
   krok:=krokanim+1;
   if strona=1 then kolor:=2 else kolor:=1;
   obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
  end;
 end;
end;

procedure animacja;
var
i,j:integer;
begin
 krok:=0;
 for i:=1 to mkrok do if swietl[i]=2 then krok:=i;
 if krok<>0 then
 begin
  krokanim:=krok;
  animmax:=krok+animile;
  if animmax>mkrok then animmax:=mkrok;
  for j:=krok to animmax-1 do
  begin
   krok:=j;
   fillchar(walka[mkrok+1]^.w[1,1,1],6*mstrona*mgrupa*mblok,0);
```

```
 move(walka[a]^.w[strona,grupa,1],walka[b]^.w[strona,grupa,1],2*srodek-2);
 for i:=1 to walka[b]^.w[strona,grupa,ilosc] do
  move(walka[c]^.w[strona,grupa,srodek],walka[b]^.w[strona,grupa,srodek-2+2*
end;

procedure dajanim;
var
i,ksz1,ksz2,ile1,ile2:integer;
begin
 ksz1:=walka[krokanim]^.w[strona,grupa,ksztalt];
 ksz2:=walka[krokanim+1]^.w[strona,grupa,ksztalt];
 if ((ksz1<>0) or (ksz2<>0)) then
 begin
  ile1:=walka[krokanim]^.w[strona,grupa,ilosc];
  ile2:=walka[krokanim+1]^.w[strona,grupa,ilosc];
  if ksz1=ksz2 then
  begin
   if ile1=ile2 then daj(krokanim,mkrok+1);
   if ile1<ile2 then dodaj(krokanim,mkrok+1,ile2,ile1);
   if ile1>ile2 then
   begin
    daj(krokanim,mkrok+2);
    dodaj(krokanim+1,mkrok+3,ile1,ile2);
   end;
  end
  else
  begin
   if ksz1=0 then punkt(krokanim+1,mkrok+1,krokanim+1)
   else
   begin
    if ksz2=0 then
    begin
     daj(krokanim,mkrok+2);
     punkt(krokanim,mkrok+3,krokanim);
    end
    else
    begin
     punkt(krokanim+1,mkrok+1,krokanim);
     daj(krokanim,mkrok+2);
     punkt(krokanim,mkrok+3,krokanim);
    end;
   end;
  end;
 end;
end;

{zmien i wyrysuj obrazek animacji dazac od kroku a do b}
procedure zmien(a,b:integer);
var
i:integer;
begin
 for i:=0 to 2*walka[a]^.w[strona,grupa,ilosc]-1 do
  walka[a]^.w[strona,grupa,srodek+i]:=
   walka[a]^.w[strona,grupa,srodek+i]+
    (walka[b]^.w[strona,grupa,srodek+i]-
     walka[a]^.w[strona,grupa,srodek+i]) div
      (animstep-astep+1);
 if strona=1 then kolor:=2 else kolor:=1;
 krok:=a;
 obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
```

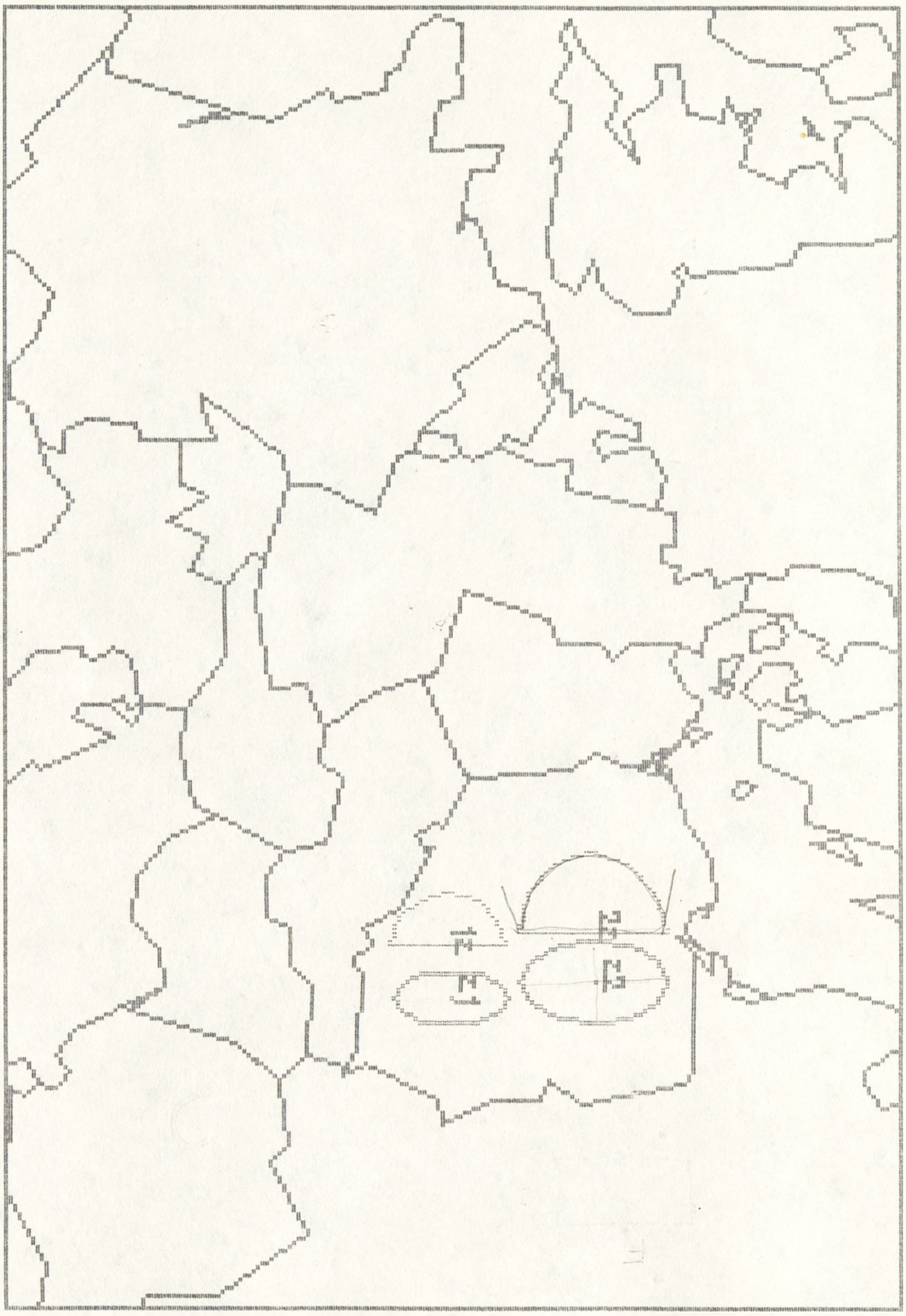
```
    begin
    cyfra(11);
    end;
    if numer>9 then
    begin
     cyfra((numer div 10)+1);
     numer:=numer mod 10;
    end;
    cyfra(numer+1);
    if strona=2 then cyfra(11);
   end;
  end;
 end;
end;

procedure oznakuj;
begin
 if lad=1 then oznakujo(minl,maxl);
 if powietrze=1 then oznakujo(minp,maxp);
 if woda=1 then oznakujo(minw,maxw);
end;

{****}


{**** animacja ****}

{zmienne:}
var
animile:integer;{ilosc krokow do animacji}
animstep:integer;{ilosc animacji wkroku}
astep:integer;{aktualna animacja w kroku}
animmax:integer;{max nr kroku do biezacej animacji}
krokanim:integer;{schowanie kroku animacji}
dzien,godz:integer;
czas:real;
const
godzkrok:integer=24;
break:integer=500;

{obrazek z a daj do b}
procedure daj(a,b:integer);
begin
 move(walka[a]^.w[strona,grupa,1],walka[b]^.w[strona,grupa,1],2*mblok);
end;

{obrazek a daj do b i dodaj c-d punktow na koncu}
procedure dodaj(a,b,c,d:integer);
var i:integer;
begin
 daj(a,b);
 walka[b]^.w[strona,grupa,ilosc]:=walka[a]^.w[strona,grupa,ilosc];
 for i:=0 to (c-d-1) do
 move(walka[b]^.w[strona,grupa,srodek+2*d-2],
  walka[b]^.w[strona,grupa,srodek+2*(d+i)],4);
end;

{obrazek ksztaltu a daj do b zageszczajac do srodka c}
procedure punkt(a,b,c:integer);
begin
```

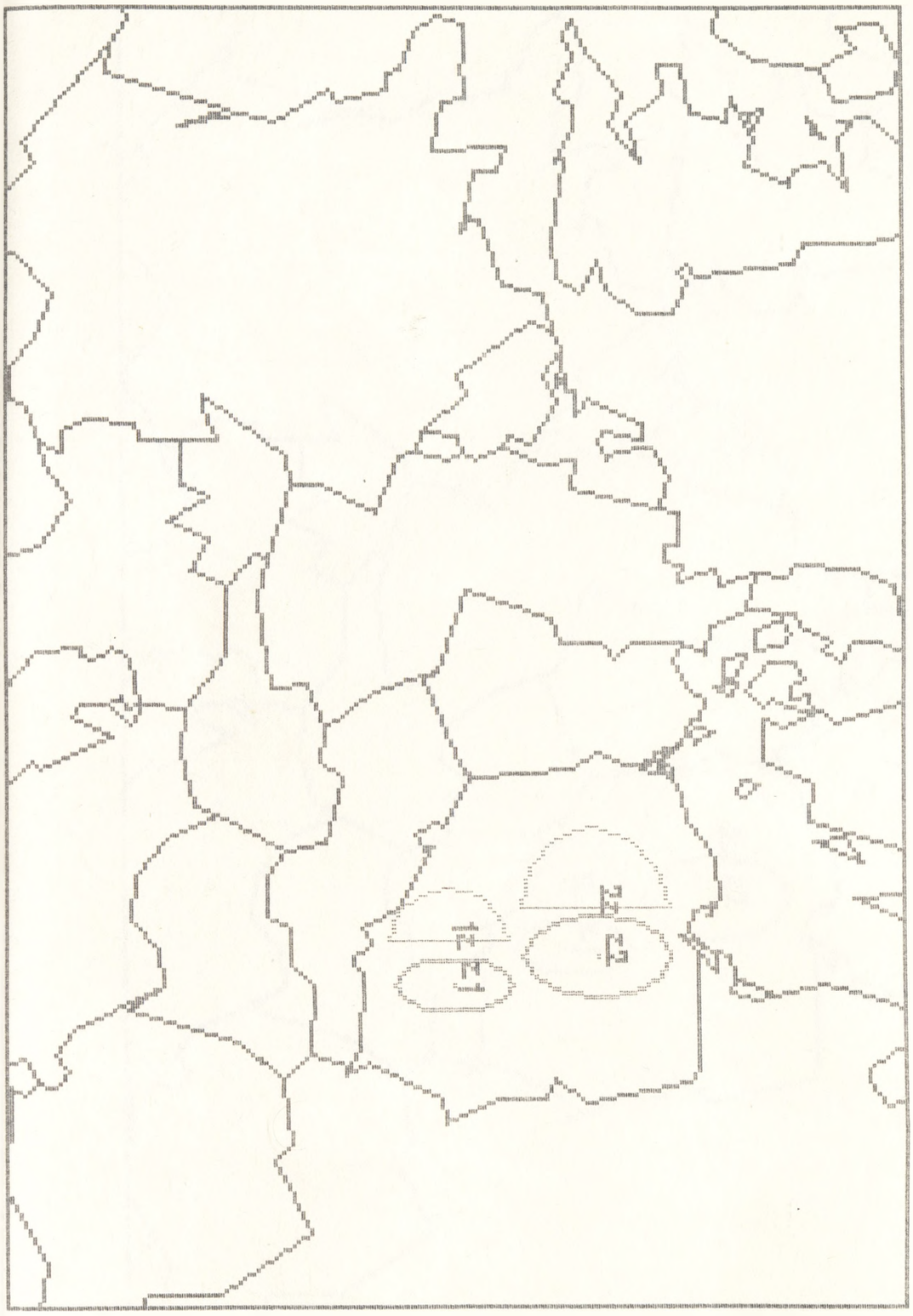
```
  if strona=1 then kolor:=2 else kolor:=1;
  if swietl[krok]=1 then kolor:=0;
  if lad=1 then for grupa:=minl to maxl do
   obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
  if powietrze=1 then for grupa:=minp to maxp do
   obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
  if woda=1 then for grupa:=minw to maxw do
   obrazek(walka[krok]^.w[strona,grupa,ksztalt]);
 end;
end;


{**** rysowanie numerow zgrupowan ****}

procedure oznakujo(min,max:integer);
const
{tablica do generowania znakow w kolejnosci 0123456789Z}
znak:array[1..11,1..15] of integer
     =((5,0,0,2,0,2,4,0,4,0,0,0,0,0,0),
       (3,1,1,2,0,2,5,0,0,0,0,0,0,0,0),
       (6,0,1,0,0,2,0,2,2,0,4,3,4,0,0),
       (7,0,0,2,0,2,2,1,2,2,2,2,4,-1,4),
       (5,0,0,0,2,2,2,2,1,2,5,0,0,0,0),
       (6,2,0,0,0,0,2,2,2,2,4,-1,4,0,0),
       (6,2,0,0,0,0,4,2,4,2,2,0,2,0,0),
       (5,0,0,2,0,2,1,0,3,0,5,0,0,0,0),
       (7,2,2,0,2,0,0,2,0,2,4,0,4,0,2),
       (6,2,2,0,2,0,0,2,0,2,4,0,4,0,0),
       (5,0,1,2,1,0,3,0,4,3,4,0,0,0,0));
var k,x,y,numer :integer;
procedure cyfra(i:integer);
var j:integer;
begin
 x2:=x+znak[i,2];
 y2:=y+znak[i,3];
 for j:=2 to znak[i,1] do
 begin
  x1:=x2;
  y1:=y2;
  x2:=x+znak[i,j+j];
  y2:=y+znak[i,j+j+1];
  draw(x1,y1,x2,y2,kolor);
 end;
 x:=x+4;
end;
{}
begin
  for strona:=1 to 2 do
  begin
  for k:=min to max do
  begin
   numer:=walka[krok]^.w[strona,k,ugrup];
   if ((numer>0) and (numer<100)) then
   begin
    x:=walka[krok]^.w[strona,k,srodek];
    y:=walka[krok]^.w[strona,k,srodek+1];
    zmiana(x,y);
    x:=x-5;
    y:=y-6;
    if strona=1 then
```

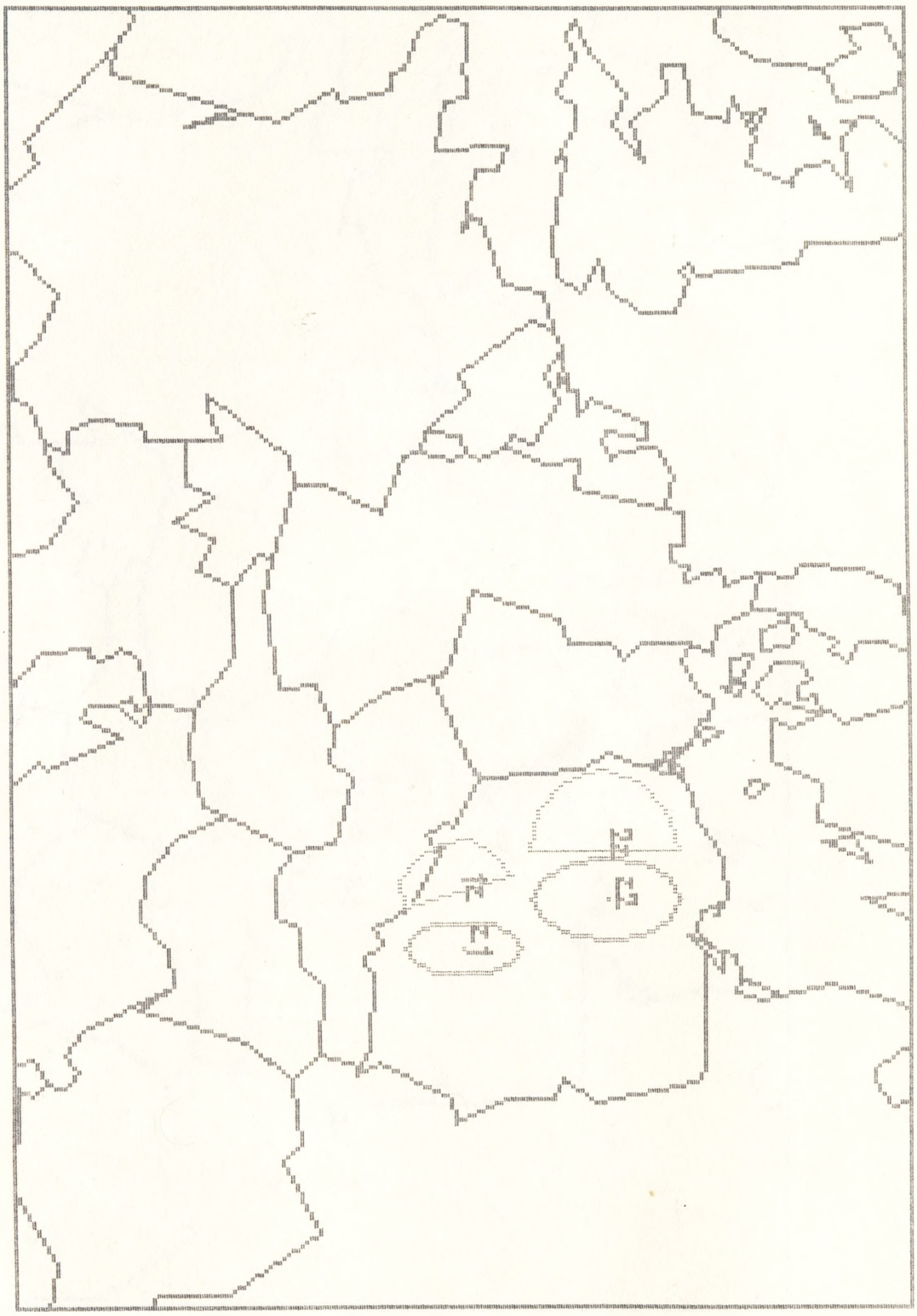
```
 3:plotko(x1,y1);
 4:prlok(x1,y1);
 end;
end;

const
tabtyp:array[1..16] of integer
=(1,2,1,1,1,2,2,2,2,2,2,2,1,2,1,2);
{1=obrazek zamkniety,2=obrazek otwarty}
tabpismo:array[1..16] of integer
=(1,1,1,1,1,1,2,3,4,5,1,6,6,1,6,1);
tabzntak:array[1..16] of integer
=(1,1,1,1,1,1,1,1,1,1,2,2,2,3,3,4);
{rysowanie obrazka ugrupowania}
procedure obrazek(ksztalt:integer);
var
i,j,typ,pismo,zntak:integer;
begin
 if ksztalt>0 then
 begin
  typ:=tabtyp[ksztalt];
  pismo:=tabpismo[ksztalt];
  zntak:=tabzntak[ksztalt];
  i:=walka[krok]^.w[strona,grupa,ilosc];
  x1:=walka[krok]^.w[strona,grupa,srodek];
  y1:=walka[krok]^.w[strona,grupa,srodek+1];
  zmiana(x1,y1);
  if i=1 then zntakt(zntak,x1,y1+3) else
  begin
   plot(x1,y1,kolor);
   x1:=walka[krok]^.w[strona,grupa,srodek+2];
   y1:=walka[krok]^.w[strona,grupa,srodek+3];
   zmiana(x1,y1);
   zntakt(zntak,x1,y1);
   for j:=1 to i-2 do
   begin
    x2:=walka[krok]^.w[strona,grupa,j+j+srodek+2];
    y2:=walka[krok]^.w[strona,grupa,j+j+srodek+3];
    zmiana(x2,y2);
    kreslic(pismo);
    zntakt(zntak,x2,y2);
    x1:=x2;
    y1:=y2;
   end;
   if typ=1 then
   begin
    x2:=walka[krok]^.w[strona,grupa,srodek+2];
    y2:=walka[krok]^.w[strona,grupa,srodek+3];
    zmiana(x2,y2);
    kreslic(pismo);
   end;
  end;
 end;
end;

{rysowanie sytuacji bojowej}
procedure sytuacja;
begin
 for strona:=1 to mstrona do
 begin
```

```
program animacja;


const
mkrok=10;
mstrona=2;
mgrupa=99;
mblok=40;
mkroki=13;
nasi=1;
oni=2;

type
stos1=^tabl1;
tabl1=record
    w: array[1..mstrona,1..mgrupa,1..mblok] of integer;
    end;
var
walka :array[1..mkroki] of stos1;
{ blok:1-numer,2-ksztalt,3-F,4-G,5-kat,6-ilosc,7'8-srodek}
krok,strona,grupa,blok : integer;

swietl : array[1..mkrok] of integer;
{1-nie,2-wyswietlone}
lad,powietrze,woda:integer;
const
minl:integer=1;
maxl:integer=49;
minp:integer=50;
maxp:integer=79;
minw:integer=80;
maxw:integer=99;


{****      procedury     ****}

procedure zmiana(var x,y:integer);
begin
 x:=round(skalax*(x-xpocz));
 y:=round(skalay*(y-ypocz));
end;


{**** rysowanie sytuacji  ****}


{kreslenie znaku takt lub punkt}
procedure zntakt(zntak,x1,y1:integer);
begin
 case zntak of
 1:plot(x1,y1,kolor);
 2:pdywrak(x1,y1);
```

PRZYKŁADOWE WYDRUKI PROBLEMU ANIMACJI






E2
E3
E4
E1