Autor: Muntean Paul Ioan

Profesor coordonator: Borşia Călin

Teiuş 2006

Introducere

Acest soft conţine rezolvarea unor problemelor de informatică care sunt mai deosebite decat cele intalnite in manuale.
El vine în ajutorul elevilor de liceu cu scopul de a prezenta o alta latura a programari in pascal ,acesta se adreseaza in mod special a celor ce sunt impatimiti de programare si sunt dispusi sa petreaca multe ore la elaborarea softurilor.

Programul este realizat în limbajul de programare TMT Pascal v3.90 şi este conceput sub forma unui corp principal a programului de unde se apeleaza fiecare program si un unit unde se gasec toate problemele inmagazinatei,asemenea unei centrale care conduce totul,programul meu avand un grad mare de modularizare in idea de a fi mai transparent si mai usor de intretinut .

Interfaţa prietenoasă a acestui program este sub forma unui meniu ce oferă utilizatorului posibilitatea de a alege problema dorită prin simplu click cu mouse-ul deasupra numarului din fata fiecarui meniu,fiecarei probleme.Fiecare numar este legat de o anumita optiune ce se va executa in urma parcurgeri meniului.
Pentru a parasi fiecare meniu se apasa pe butonl "EXIT" din fiecare meniu.

Pentru ca programul sa samene cat mai mult cu cele din zilele noastre sa folositt efecte din uniturile:CRT ,keyboard,DOS, graph,dos,ATES,{$ifdef__WIN32__}Windows,MMsystem ,GRAPH {$endif} ;
Softul se instaleaza cu ajutorul unui chit de instalare numit Paul MUNTEAN IOAN.EXE.

Pentru buna funcţionare a acestui program este recomandată folosirea unui calculator care să conţină un sistem de operare Windows Xp sau si modele mai

vechi , sa fie instalată aplicaţia TMT Pascal şi un spaţiu pe hard-disk de minim 1Mb.
Pentru o mai buna intelegere a softului se recomnda rularea lui de cate ori este nevoie.

Softul este însoţit de prezenta documentaţiei care cuprinde o scurtă introducere, textul problemelor rezolvate, codul sursă al acestora.

Cap 1.PROGRAMUL PRINCIPAL

CONDUCE INTREG PROGRAMUL:

```
    Program atestat;
uses  dos,ATES,
 {$ifdef __WIN32__} Windows, MMsystem,GRAPH    {$endif}  crt;
type
     mor=record
         cod:string;
         carac:char;
         end;
     natural=0..maxlongint;
  var v1:array[1..46] of mor;
      s4:string;
      st:array[1..100] of integer;
      opt1,opt:integer;
      f:text;
      g,k:integer;
      val:integer;
      x,y,z,w,r,p:integer;
      h3,m3,s3,s1003,h,m,s,s100,h2,h1,m2,m1,s1,s2,s1002,s1001,a,c,d:word;
      j:integer;

begin
 textbackground(blue); clrscr;
TEXTCOLOR(YELLOW);gotoxy(7,12);hidecursor;  MouseInit;
```

```
Write('MENIURILE SE POT ACCESA PRIN CLICK PE NUMARUL DIN
FATA FIECARUI MENIU');DELAY(5000);clrscr;mousehide;
repeat
 textbackground(blue);  clrscr;

   {a:=h;b:=m;c:=s;d:=s100; }
   textcolor(white);gotoxy(62,23);writeln('Start   ',h1,':',m1,':',s1,':',s1001);
{ a:=h;b:=m;c:=s;d:=s100;} gotoxy(62,24);writeln('Sfarsit
',h2,':',m2,':',s2,':',s1002);

GOTOXY(2,2); mousedata(mouse,x,y);  MouseInit;
write('Paul-Ioan Muntean alias MASTERCODE&CO');

gotoxy(25,7);textcolor(yellow);
writeln('1.');textbackground(blue);gotoxy(27,7);write('Meniul
grafic');gotoxy(25,8);
writeln('2.');textbackground(blue);gotoxy(27,8);write('Meniul cu altgoritmi
matematici ');gotoxy(25,9);
writeln('3.');textbackground(blue);gotoxy(27,9);write('Meniul cu
fisiere');gotoxy(25,10);
writeln('4.');textbackground(blue);gotoxy(27,10);write('Meniul cu
grafuri');gotoxy(25,11);
writeln('5.');textbackground(blue);gotoxy(27,11);write('Meniul
BKTR');gotoxy(25,12);
writeln('6.');textbackground(blue);gotoxy(27,12);write('Meniul Divide et
Impera');gotoxy(25,13);
writeln('7.');textbackground(blue);gotoxy(27,13);write('Meniul cu altgoritmi
speciali');gotoxy(25,14);
writeln('8.');textbackground(blue);gotoxy(27,14);write('Meniul cu stringuri');
gotoxy(33,17);textbackground(green);
writeln('EXIT');gotoxy(29,6);  textbackground(blue);
 textcolor(white);write('*OPTIUNEA TA ESTE*'); textcolor(yellow);
gotoxy(27,14);write('Meniul cu stringuri');
                  repeat
                 gotoxy(25,7);textcolor(yellow);writeln('1.');


{ in repeatul acesta ii easul din stnga sus integrat si functinea mouseului}
```

```
mousedata(mouse,x,y);
gettime(h3,m3,s3,s1003);textcolor(white);gotoxy(62,2);textbackground(red);wri
te('Time ',h3,':',m3,':',s3,':',s1003);textcolor(yellow);gotoxy(44,16);
hidecursor;  mousedata(mouse,x,y);  MouseShow;textbackground(blue);
gotoxy(34,14);write('cu s'); textbackground(cyan);


{1}if ((mouse=1)and((x=25)or(x=26))) and( y=7 ) then opt:=1;
{2}if(mouse=1)and((x=25)or(x=26))and( y=8 ) then opt:=2;
  if(mouse=1)and (x=25) and( y=9 ) then opt:=3;
  if(mouse=1)and (x=25) and( y=10 )then opt:=4;
   if(mouse=1)and (x=25) and( y=11 ) then opt:=5;
    if(mouse=1)and (x=25) and( y=12 ) then opt:=6;
     if (mouse=1)and(x=25) and( y=13 ) then opt:=7;
     if(mouse=1)and (x=25) and( y=14 ) then opt:=8;
      if (mouse=1)and((x=33) or (x=34) or (x=35)or (x=36)) and( y=17 ) then exit
                                else
                                 begin
                                if(((x=25)or(x=26)) and(y=7))  then  Begin
gotoxy(25,7);textbackground(red); end;
                                if not((x=25)or(x=26)) and (y=7) then begin
gotoxy(25,7);  textbackground(cyan);      end;
                                end;
until (mouse=1) and ((x in [25..30])and (y in[7..17]))  ;
 mousehide;
clrscr;nosound;
case opt of
1:begin
  repeat
          textbackground(blue); clrscr;    mousehide;   mouseinit;
MOUSESHOW;
             gotoxy(30,7);     mousedata(mouse,x,y);
     write('1.');textbackground(blue);gotoxy(32,7);write('Ferestre
aleatoare');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Afisarea aleatoare a
caracterelor ASCII');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Afisarea  creatorului
programului');gotoxy(30,10);
```

```
    write('4.');textbackground(blue);gotoxy(32,10);write('Matrix Revolution&
Machine cod');gotoxy(30,11);
    write('5.');textbackground(blue);gotoxy(32,11);write('Triunghi de numere
sonorizat de note muzicale');
    gotoxy(34,14);  textbackground(green);
      writeln('EXIT');
   textbackground(blue);
     repeat          gotoxy(30,7);
    write('1.');  mousedata(mouse,x,y);
        if ((mouse=1)and((x=30)or(x=31)) and( y=7 )) then opt1:=1;
         if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
           if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
             if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
               if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
               if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                  else
                               begin
                               if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                               if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);     end;

                               end;
   until (mouse=1 )and (x in[30..37])and (y in [7..17])    ;
     case opt1 of
     1:begin
    gettime(h1,m1,s1,s1001);
   CrtDemo;                          {MERGE FARA ERORI DACA SUNT
INCAPSULATE}
    gettime(h2,m2,s2,s1002);
     end;
    2:begin
       gettime(h1,m1,s1,s1001);clrscr;
        gr1;
       gettime(h2,m2,s2,s1002);
      end;
 3:begin
  gettime(h1,m1,s1,s1001);clrscr;
```

```
        gr2;
       gettime(h2,m2,s2,s1002);
end;
 4:begin
  gettime(h1,m1,s1,s1001);clrscr;
        gr3;
       gettime(h2,m2,s2,s1002);
end;
   5:begin
   gettime(h1,m1,s1,s1001);clrscr;
        gr4;
       gettime(h2,m2,s2,s1002);
     end;

   end;
  until opt1=0; clrscr;  textbackground(blue); clrscr;
  end;
2:begin
 repeat
          textbackground(blue);clrscr;     MOUSESHOW;
            gotoxy(30,7);     mousedata(mouse,x,y);hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Ec. de grad 2 rezolvata
in C ');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Bordarea unei matrici
cu 1 coloana si 1 linie');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Transormarea unui nr.
in orice baza');gotoxy(30,10);
     write('4.');textbackground(blue);gotoxy(32,10);write('Inmultirea a 2
matrici');gotoxy(30,11);

write('5.');textbackground(blue);gotoxy(32,11);write('Ackerman');gotoxy(30,12
);
     write('6.');textbackground(blue);gotoxy(32,12);write('Mana_Pnuelli; ');
gotoxy(34,14);
    textbackground(green); write('EXIT');
      repeat          gotoxy(30,7);     mousedata(mouse,x,y);
     write('1.');  mousedata(mouse,x,y);
         if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
          if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
```

```
            if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
              if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
                if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                   if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
                if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                              else
                                  begin
                                  if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                  if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                  end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
      case opt1 of
      1:begin
 gettime(h1,m1,s1,s1001);
        mat1;
  gettime(h2,m2,s2,s1002);
       end;
      2:begin
      gettime(h1,m1,s1,s1001);
      mat2;
        gettime(h2,m2,s2,s1002);
      end;
        3:begin
         gettime(h1,m1,s1,s1001);
      mat3;
        gettime(h2,m2,s2,s1002);
      end;
      4:begin
          gettime(h1,m1,s1,s1001);
      mat4;
        gettime(h2,m2,s2,s1002);
    end
       5:begin
        gettime(h1,m1,s1,s1001);
         functia_Ackerman;clrscr;
        gettime(h2,m2,s2,s1002);
```

```
       end;
       6:begin
        gettime(h1,m1,s1,s1001);
          Mana_Pnuelli; clrscr;
        gettime(h2,m2,s2,s1002);
       end;
      end;
      until opt1=0;
end;
 3:begin
  repeat
      textbackground(blue); clrscr;MouseInit;   MOUSESHOW;
            gotoxy(30,7);    mousedata(mouse,x,y); textcolor(yellow);hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Ordonarea alfabetica a
unei liste din fisier');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Simulare click
2');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Se genereaza parola in
Parola.doc');gotoxy(30,10);
     write('4.');textbackground(blue);gotoxy(32,10);write('Umplere fisier cu
caractere');gotoxy(30,11);
     write('5.');textbackground(blue);gotoxy(32,11);write('ARNOLD
SCHWARZENEGER');gotoxy(30,12);
     write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
    textbackground(green); write('EXIT');   textbackground(blue);
      repeat          gotoxy(30,7);     mousedata(mouse,x,y);
     write('1.');  mousedata(mouse,x,y);   mousedata(mouse,x,y);
         if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
          if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
            if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
              if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
                if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                  if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
                if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                        else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
```

```
                              if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                              end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
     case opt1 of
     1:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
       write('Se prelucreaza');   readln;textcolor(yellow);   Ara;
        gettime(h2,m2,s2,s1002);
      end;
     2:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
         click2;textcolor(yellow);
        gettime(h2,m2,s2,s1002);
      end;
     3:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
          genParola;
        gettime(h2,m2,s2,s1002);
     end;
     4:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
         umplere_fisier_text;
        gettime(h2,m2,s2,s1002);
      end;
     5:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
        arnold;
        gettime(h2,m2,s2,s1002);
      end;
     0:begin
       gettime(h1,m1,s1,s1001);CLRSCR;
        gettime(h2,m2,s2,s1002);
      end
      else
        begin
      write('Dati o optiune din 1..5 sau 0');
        end;
     end;
```

```
      until opt1=0;
      end;
4:begin
  repeat
    textbackground(blue); clrscr; MouseInit;   MOUSESHOW;
        gotoxy(30,7);    mousedata(mouse,x,y); textcolor(yellow);  hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Verifica daca un graf
este sau nu hamiltonian');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Componenta tare
conexa');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Eliminarea unui varf
a.i. graful sa fie conex');gotoxy(30,10);
     write('4.');textbackground(blue);gotoxy(32,10);write('Verifica daca un graf
este eulerian si conex');gotoxy(30,11);
     write('5.');textbackground(blue);gotoxy(32,11);write('Afisare arborelui
partial de cost minim');gotoxy(30,12);
     write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
    textbackground(green); write('EXIT');   textbackground(blue);
       repeat          gotoxy(30,7);     mousedata(mouse,x,y);
     write('1.');  mousedata(mouse,x,y);  mousedata(mouse,x,y);
         if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
          if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
            if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
              if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
                if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                   if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
                if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                        else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
      case opt1 of
     1:begin
       gettime(h1,m1,s1,s1001);CLRSCR;
```

```
          graf_hamiltonian;readln;
         gettime(h2,m2,s2,s1002);
        end;
       2:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
          componenta_tareconex;
         gettime(h2,m2,s2,s1002);
       end;
       3:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
           subGrafConex;
         gettime(h2,m2,s2,s1002);
       end;
       4:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
          grafEulerian;
         gettime(h2,m2,s2,s1002);
        end;
       5:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
           arb_partial_de_cost_minim;
         gettime(h2,m2,s2,s1002);
        end;
       0:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
          gettime(h2,m2,s2,s1002);
        end
        else
          begin
        write('Dati o optiune din 1..5 sau 0');
          end;
       end;
       until opt1=0;
       end;
  5:begin
    repeat
     textbackground(blue); clrscr;  MouseInit;   MOUSESHOW;
           gotoxy(30,7);   mousedata(mouse,x,y); textcolor(yellow);    hidecursor;
```

```
    write('1.');textbackground(blue);gotoxy(32,7);write('Afisarea permutarilor
de n elemente');gotoxy(30,8);
    write('2.');textbackground(blue);gotoxy(32,8);write('Afisarea combinarilor
de n,k elemente');gotoxy(30,9);
    write('3.');textbackground(blue);gotoxy(32,9);write('Afisare produs
cartezian de n multimi');gotoxy(30,10);
    write('4.');textbackground(blue);gotoxy(32,10);write('Generarea nr. prime
din n cifre');gotoxy(30,11);
    write('5.');textbackground(blue);gotoxy(32,11);write('Comis
voiajor');gotoxy(30,12);
    write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
   textbackground(green); write('EXIT');   textbackground(blue);
     repeat          gotoxy(30,7);     mousedata(mouse,x,y);
    write('1.');  mousedata(mouse,x,y);   mousedata(mouse,x,y);
        if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
         if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
          if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
            if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
              if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                 if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
              if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                        else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
     case opt1 of
     1:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
          bktr_nerecursiv_permutari;
        gettime(h2,m2,s2,s1002);
       end;
     2:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
        bktr_combinari;
```

```
        gettime(h2,m2,s2,s1002);
       end;
      3:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
         bktr_prodCartezian;
          gettime(h2,m2,s2,s1002);
      end;
      4:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
           bktr_Nrprime;
         gettime(h2,m2,s2,s1002);
       end;
      5:begin
          gettime(h1,m1,s1,s1001);CLRSCR;
           bktr_comisVOIAJOR;
         gettime(h2,m2,s2,s1002);
       end;
      0:begin
        gettime(h1,m1,s1,s1001); CLRSCR;
         gettime(h2,m2,s2,s1002);
       end
       else
         begin
       write('Dati o optiune din 1..5 sau 0');
         end;
      end;
      until opt1=0;
      end;
6:begin
  repeat
    textbackground(blue); clrscr;MouseInit;   MOUSESHOW;
        gotoxy(30,7);     mousedata(mouse,x,y); textcolor(yellow);  hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Cautare binara intr-un
sir');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('TURNURILE DIN
HANOI');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('CMMDC de N
elemente mai mari decat "0"');gotoxy(30,10);
```

```
    write('4.');textbackground(blue);gotoxy(32,10);write('Aria maxima dintr-o
foaie de tabla');gotoxy(30,11);
    write('5.');textbackground(blue);gotoxy(32,11);write('Quick sort de n
elemente');gotoxy(30,12);
    write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
   textbackground(green); write('EXIT');   textbackground(blue);
      repeat          gotoxy(30,7);     mousedata(mouse,x,y);
    write('1.');  mousedata(mouse,x,y); mousedata(mouse,x,y);
        if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
         if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
          if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
            if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
              if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                 if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
              if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                        else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
      case opt1 of
     1:begin
       gettime(h1,m1,s1,s1001); CLRSCR;
          cautare_binara;
       gettime(h2,m2,s2,s1002);
      end;
     2:begin
       gettime(h1,m1,s1,s1001); CLRSCR;
       turnurile_hanoi;
       gettime(h2,m2,s2,s1002);
     end;
     3:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
          CmmdcDEnNR;
       gettime(h2,m2,s2,s1002);
```

```
      end;
      4:begin
         gettime(h1,m1,s1,s1001); CLRSCR;
        foaiatabla;
        gettime(h2,m2,s2,s1002);
       end;
      5:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
        quick_sort;
        gettime(h2,m2,s2,s1002);
       end;
      0:begin
       gettime(h1,m1,s1,s1001); CLRSCR;
        gettime(h2,m2,s2,s1002);
       end
       else
        begin
       write('Dati o optiune din 1..5 sau 0');
        end;
      end;
      until opt1=0;
      end;
7:begin
  repeat
     textbackground(blue); clrscr;MouseInit;   MOUSESHOW;;hidecursor;
           gotoxy(30,7);     mousedata(mouse,x,y); textcolor(yellow);   hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Flame
,Flacara');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Desenare
aleatoare');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Joc
Snake');gotoxy(30,10);
     write('4.');textbackground(blue);gotoxy(32,10);write('Cod
morse');gotoxy(30,11);

write('5.');textbackground(blue);gotoxy(32,11);write('Hexagon');gotoxy(30,12);
     write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
    textbackground(green); write('EXIT');   textbackground(blue);
     repeat          gotoxy(30,7);     mousedata(mouse,x,y);
```

```
    write('1.');  mousedata(mouse,x,y);    mousedata(mouse,x,y);
        if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
         if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
          if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
            if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
              if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                 if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
              if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                        else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
       case opt1 of
     1:begin
       gettime(h1,m1,s1,s1001);CLRSCR;
       flame;
       gettime(h2,m2,s2,s1002);
      end;
     2:begin
       gettime(h1,m1,s1,s1001);CLRSCR;
       desenare;
       gettime(h2,m2,s2,s1002);
     end;
     3:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
        snake;
       gettime(h2,m2,s2,s1002);
     end;
     4:begin
        gettime(h1,m1,s1,s1001); CLRSCR;
        morse;
       gettime(h2,m2,s2,s1002);
      end;
     5:begin
```

```
          gettime(h1,m1,s1,s1001); CLRSCR;
            hexagon3D;
          gettime(h2,m2,s2,s1002);
        end;
       0:begin
         gettime(h1,m1,s1,s1001); CLRSCR;
          gettime(h2,m2,s2,s1002);
        end
        else
          begin
        write('Dati o optiune din 1..5 sau 0');
          end;
       end;
       until opt1=0;
       end;
   8:begin
     repeat
          textbackground(blue); clrscr;MouseInit;   MOUSESHOW;
            gotoxy(30,7);     mousedata(mouse,x,y); textcolor(yellow);   hidecursor;
     write('1.');textbackground(blue);gotoxy(32,7);write('Eliminarea literelor ce
se repeta inr-un cuvant');gotoxy(30,8);
     write('2.');textbackground(blue);gotoxy(32,8);write('Alcatuirea unei parole
dintr-un cuvant dat');gotoxy(30,9);
     write('3.');textbackground(blue);gotoxy(32,9);write('Sortarea alfabetica a
unei fraze');gotoxy(30,10);
     write('4.');textbackground(blue);gotoxy(32,10);write('Afisarea  inversa a
cifrelor dintr-un cuvant');gotoxy(30,11);
     write('5.');textbackground(blue);gotoxy(32,11);write('Prefixurile unui
cuvant');gotoxy(30,12);
     write('');textbackground(blue);gotoxy(32,12);write(' '); gotoxy(34,14);
    textbackground(green); write('EXIT');   textbackground(blue);
      repeat          gotoxy(30,7);     mousedata(mouse,x,y);
     write('1.');  mousedata(mouse,x,y);    mousedata(mouse,x,y);
         if ((mouse=1)and((x=30)or(x=31))) and( y=7 ) then opt1:=1;
          if(mouse=1)and((x=30)or(x=31))and( y=8 ) then opt1:=2;
            if(mouse=1)and ((x=30)or(x=31)) and( y=9 ) then opt1:=3;
              if(mouse=1)and ((x=30)or(x=31)) and( y=10 )then opt1:=4;
                if(mouse=1)and ((x=30)or(x=31)) and( y=11 ) then opt1:=5;
                  if(mouse=1)and ((x=30)or(x=31)) and( y=12 )then opt1:=6;
```

```
                if (mouse=1)and((x=34) or (x=35) or (x=36)or (x=37)) and( y=14 )
then  opt1:=0
                                                else
                                begin
                                if(((x=30)or(x=31)) and(y=7))  then  Begin
gotoxy(30,7);textbackground(red); end;
                                if ((x<>30)or(x<>31)) and (y<>7) then begin
gotoxy(30,7);  textbackground(cyan);      end;
                                end;
   until (mouse=1 )and (x in[30..37])and (y in [7..14])    ;
     case opt1 of
     1:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
         eliminareLITrepetate;
        gettime(h2,m2,s2,s1002);
       end;
     2:begin
        gettime(h1,m1,s1,s1001);CLRSCR;
         parola;
        gettime(h2,m2,s2,s1002);
      end;
     3:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
           p7;
        gettime(h2,m2,s2,s1002);
     end;
     4:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
        cuvant;
        gettime(h2,m2,s2,s1002);
       end;
     5:begin
         gettime(h1,m1,s1,s1001);CLRSCR;
          prefix;
        gettime(h2,m2,s2,s1002);
       end;
     0:begin
       gettime(h1,m1,s1,s1001); CLRSCR;
        gettime(h2,m2,s2,s1002);
```

```
        end
        else
          begin
        write('Dati o optiune din 1..5 sau 0');
          end;
      end;
      until opt1=0;
      end;
0:begin
     gettime(h1,m1,s1,s1001);
  exit;
   gettime(h2,m2,s2,s1002);
  end
  else
    begin
      gettime(h1,m1,s1,s1001); MouseInit;   MOUSESHOW;
     writeln('Alegeti optiuni intre 1..8,0 exit');
  readln; gettime(h2,m2,s2,s1002);clrscr;
    end;
 end;
until opt=0; clrscr;
end.
```

Cap 2.UNITUL ATES

CONTINE TOATE PROGRAMELE:

Procedurile pentru mouse:

```
procedure MouseInit;assembler;
  asm mov ax,0;int $33 end;
procedure MouseShow;assembler;
 asm mov ax,1; int $33 end;
procedure MouseData(var buton,x,y:integer);
```

```
var bb,xx,yy:word;
  begin
  asm mov ax,3; int $33; mov bb,bx; mov xx,cx; mov yy,dx end;
  buton:=bb;x:=xx div 8 +1;y:=yy div 8+1;
  end;
procedure MouseHide;assembler;
asm mov ax,2; int $33 end;
```

Problema 1. Meniul grafic

Codul sursă:

Afiseaza aleator ferestre si efectueaza diferite prelucrari asupra lor ;
procedure CrtDemo;

ClrScr
DelLine
GoToXY
InsLine
KeyPressed
ReadKey
TextBackground
TextColor
TextMode
WhereX
WhereY
Window
Write
WriteLn;

Also uses LastMode and WindMax variables from Crt unit.

1. Init routine:
- Save original video mode. On an EGA or VGA, use the 8x8 font (43 lines on an EGA, 50 on VGA).

- Setup LastRow to preserve last line on screen for messages (preserves last 2 lines in 40-column mode). Setup LastCol.
- Initialize the random number generator.

2. MakeWindow routine:
   - Puts up random-sized, random-colored windows on screen.
3. Program body:
   - Call Init
   - Loop until Contrl-C is typed:
     - Echo keystrokes (Turbo Pascal windows automatically wrap and scroll).
     - Support special keys:
       <Ins> inserts a line at the cursor
       <Del> deletes a line at the cursor
       <Up>,
       <Dn>,
       <Right>,
       <Left> position the cursor in the window
       <Alt-R> generate random text until a key is pressed
       <Alt-W> creates another random window
       <ESC> exits the program

}

```
var
  OrigMode,LastCol,LastRow: Word;
  Ch: Char;
  Done: Boolean;

procedure Initialize;
{ Initialize the video mode, LastCol, LastRow, and the random number }
{ generator. Paint the help line. }
begin
  CheckBreak:=False;                { turn off Contrl-C checking }
  OrigMode:=LastMode;               { Remember original video mode }
  TextMode(Lo(LastMode)+Font8x8);   { use 43 or 50 lines on EGA/VGA }
  LastCol:=Lo(WindMax)+1;           { get last column, row }
  LastRow:=Hi(WindMax)+1;
  GoToXY(1,LastRow);                { put message line on screen }
```

```
  TextBackground(Black);
  TextColor(White);
  Write('Ins- +linie  ',
        'Del- -linie  ',
        'Arrows-Cursor  ',
        'Alt-W-New ',
        'Alt-R-Generare  ',
        'Esc-Iesire');
  Dec(LastRow,80 div LastCol);        { don't write on message line }
  Randomize;                          { init random number generator }
end; { Init }

procedure MakeWindow;
{ Make a random window, with random background and foreground colors }
var
  X,Y,Width,Height: Word;
begin
  Width:=Random(LastCol-2)+2;             { random window size }
  Height:=Random(LastRow-2)+2;
  X:=Random(LastCol-Width)+1;          { random position on screen }
  Y:=Random(LastRow-Height)+1;
  Window(X,Y,X+Width,Y+Height);
  if OrigMode = Mono then
  begin
    TextBackground(White);
    TextColor(Black);
    ClrScr;
    Window(X+1,Y+1,X+Width-1,Y+Height-1);
    TextBackground(Black);
    TextColor(White);
    ClrScr;
  end
  else
  begin
    TextBackground(Random(8));
    TextColor(Random(7)+9);
  end;
  ClrScr;
end; { MakeWindow }
```

```
procedure RandomText;
{ Generate random text until a key is pressed. Filter out }
{ control characters. }
begin
  repeat
    Write(Chr(Random(256-32)+32));
  until KeyPressed;
end; { RandomText }

begin { program body }
  Initialize;
  MakeWindow;
  Done:=False;
  repeat
    Ch:=ReadKey;
    case Ch of
      #0:                          { Function keys }
      begin
        Ch:=ReadKey;
        case Ch of
          #17: MakeWindow;           { Alt-W  }
          #19: RandomText;           { Alt-R  }
          #45: Done:=True;           { Alt-X  }
          #72: GotoXY(WhereX,WhereY-1); { Up     }
          #75: GotoXY(WhereX-1,WhereY); { Left   }
          #77: GotoXY(WhereX+1,WhereY); { Right  }
          #80: GotoXY(WhereX,WhereY+1); { Down   }
          #82: InsLine;              { Ins    }
          #83: DelLine;              { Del    }
        end;
      end;
      #3: Done:=True;              { Ctrl-C }
      #13: WriteLn;                { Enter  }
      #27: Done:=True;             { Esc    }
    else
      Write(Ch);
    end;
  until Done;
```

```
  TextMode(OrigMode);textcolor(yellow);
end;
```

Problema 2. Meniul grafic

Codul sursă:

```
Afiseaza aleator toate caracterele ASCII cu diferite culori;
  procedure gr1;
     begin
gotoxy(12,12);
writeln('ECRANUL VA FI INUNDAT!!!');delay(3000);
clrscr;
repeat
 TEXTcolor(x); textbackground(y+blink);
x:=random(80);y:=random(80);l:=random(256);
gotoxy(x,y);write(char(l));
z:=random(80);w:=random(80);
gotoxy(z,w);write(char(l));
s:=random(80);r:=random(80);
gotoxy(s,r);write(char(p));
d:=random(80);m:=random(80);
gotoxy(d,m);write(Char(l));
until keypressed;
     clrscr;
   gotoxy(23,14);
   write('what are you thinking now?');delay(3000);
    delay(1100);clrscr;
      textbackground(red+181);
  for i:= 1 to 25 do
   begin
gotoxy(i,i);write('$ PM $');
   end;p:=0;
    for i:= 25 downto 1 do
       begin
       inc(p);
gotoxy(i,p);write('$ PM $');
       end;
gotoxy(23,12);write(' /////----------------------------------------\\\\\');
```

```
gotoxy(23,13);write('~~~~created by the smartest boy alias MASTERCODS
PM ~~~~');
gotoxy(23,14);write(' \\\\\----------------------------------------
/////');gotoxy(30,16);write('PaulMuntean@gmail.com');
    delay(5000); textbackground(blue); clrscr;textcolor(yellow);
     readln;clrscr;
      end;
```

Problema 3,Meniul grafic.

Codul sursă:

```
Afiseaza numele creatorului acestui soft;
procedure gr2;
 begin
   gettime(h1,m1,s1,s1001);clrscr;
gotoxy(12,12);  writeln ('ecranul va fi coplesit!!!!!!!!'); delay(3000);
clrscr;

repeat
randomize;

x:=random(256);
p:=random(256);
s:=random(256);
l:=random(256);
e:=random(256);
r:=random(256);
d:=random(256);
c:=random(256);
textbackground(x);
TEXTcolor(l);
TEXTcolor(r);
TEXTcolor(d);
TEXTcolor(c);
```

```
TEXTcolor(s);
TEXTcolor(e);
TEXTcolor(s);
textcolor(p);
gotoxy(6,2);write(char(p));   {P}
gotoxy(6,3);write(char(s));
gotoxy(6,4);write(char(l));
gotoxy(6,5);write(char(e));
gotoxy(6,6);write(char(r));
gotoxy(6,7);write(char(d));
gotoxy(6,8);write(char(c));
gotoxy(6,9);write(char(p));
gotoxy(7,2);write(char(p));
gotoxy(7,6);write(char(p));
gotoxy(8,2);write(char(p));
gotoxy(8,6);write(char(p));
gotoxy(9,3);write(char(l));
gotoxy(9,4);write(char(e));
gotoxy(9,5);write(char(r));
gotoxy(9,2);write(char(d));
gotoxy(9,6);write(char(s));

gotoxy(11,3);write(char(l));   {A}
gotoxy(11,4);write(char(e));
gotoxy(11,5);write(char(r));
gotoxy(11,6);write(char(d));
gotoxy(11,7);write(char(s));
gotoxy(11,8);write(char(c));
gotoxy(11,9);write(char(p));
gotoxy(12,2);write(char(p));
gotoxy(13,2);write(char(l));
gotoxy(14,3);write(char(s));
gotoxy(14,4);write(char(r));
gotoxy(14,5);write(char(p));
gotoxy(14,6);write(char(p));
gotoxy(14,7);write(char(p));
gotoxy(14,8);write(char(p));
gotoxy(14,9);write(char(d));
gotoxy(12,5);write(char(c));
```

```
gotoxy(13,5);write(char(p));
    TEXTcolor(e);
gotoxy(16,2);write(char(p));   {U}
gotoxy(16,3);write(char(x));
gotoxy(16,4);write(char(p));
gotoxy(16,5);write(char(s));
gotoxy(16,6);write(char(c));
gotoxy(16,7);write(char(d));
gotoxy(16,8);write(char(l));
gotoxy(19,2);write(char(d));
gotoxy(19,3);write(char(p));
gotoxy(19,4);write(char(p));
gotoxy(19,5);write(char(s));
gotoxy(19,6);write(char(l));
gotoxy(19,7);write(char(r));
gotoxy(19,8);write(char(d));
gotoxy(17,9);write(char(x));
gotoxy(18,9);write(char(p));
  TEXTcolor(r);
gotoxy(21,2);write(char(s));   {L}
gotoxy(21,3);write(char(l));
gotoxy(21,4);write(char(d));
gotoxy(21,5);write(char(c));
gotoxy(21,6);write(char(p));
gotoxy(21,7);write(char(p));
gotoxy(21,8);write(char(p));
gotoxy(21,9);write(char(c));
gotoxy(22,9);write(char(e));
gotoxy(23,9);write(char(p));
gotoxy(24,9);write(char(p));
  TEXTcolor(c);

gotoxy(27,2);write(char(p));   {M}
gotoxy(27,3);write(char(p));
gotoxy(27,4);write(char(p));
gotoxy(27,5);write(char(p));
gotoxy(27,6);write(char(s));
gotoxy(27,7);write(char(l));
gotoxy(27,8);write(char(p));
```

```
gotoxy(27,9);write(char(p));
gotoxy(28,3);write(char(p));
gotoxy(29,4);write(char(r));
gotoxy(30,3);write(char(d));
gotoxy(31,3);write(char(p));
gotoxy(31,2);write(char(c));
gotoxy(31,3);write(char(p));
gotoxy(31,4);write(char(p));
gotoxy(31,5);write(char(p));
gotoxy(31,6);write(char(p));
gotoxy(31,7);write(char(s));
gotoxy(31,8);write(char(l));
gotoxy(31,9);write(char(p));

gotoxy(33,2);write(char(p));   {U}
gotoxy(33,3);write(char(p));
gotoxy(33,4);write(char(p));
gotoxy(33,5);write(char(r));
gotoxy(33,6);write(char(p));
gotoxy(33,7);write(char(d));
gotoxy(33,8);write(char(l));
gotoxy(36,2);write(char(s));
gotoxy(36,3);write(char(p));
gotoxy(36,4);write(char(x));
gotoxy(36,5);write(char(p));
gotoxy(36,6);write(char(x));
gotoxy(36,7);write(char(p));
gotoxy(36,8);write(char(p));
gotoxy(34,9);write(char(p));
gotoxy(35,9);write(char(p));

gotoxy(38,2);write(char(s));   {N}
gotoxy(38,3);write(char(p));
gotoxy(38,4);write(char(p));
gotoxy(38,5);write(char(r));
gotoxy(38,6);write(char(p));
gotoxy(38,7);write(char(x));
gotoxy(38,8);write(char(p));
gotoxy(38,9);write(char(p));
```

```
gotoxy(39,3);write(char(p));
gotoxy(40,4);write(char(p));
gotoxy(41,2);write(char(p));
gotoxy(41,3);write(char(p));
gotoxy(41,4);write(char(p));
gotoxy(41,5);write(char(l));
gotoxy(41,6);write(char(p));
gotoxy(41,7);write(char(p));
gotoxy(41,8);write(char(p));
gotoxy(41,9);write(char(p));
    TEXTcolor(d);
gotoxy(45,2);write(char(p));   {T}
gotoxy(45,3);write(char(p));
gotoxy(45,4);write(char(p));
gotoxy(45,5);write(char(p));
gotoxy(45,6);write(char(p));
gotoxy(45,7);write(char(s));
gotoxy(45,8);write(char(l));
gotoxy(45,9);write(char(r));
gotoxy(43,2);write(char(d));
gotoxy(44,2);write(char(c));
gotoxy(46,2);write(char(x));
gotoxy(47,2);write(char(p));

gotoxy(49,2);write(char(p));   {E}
gotoxy(49,3);write(char(p));
gotoxy(49,4);write(char(p));
gotoxy(49,5);write(char(l));
gotoxy(49,6);write(char(p));
gotoxy(49,7);write(char(r));
gotoxy(49,8);write(char(d));
gotoxy(49,9);write(char(c));
gotoxy(50,2);write(char(s));
gotoxy(51,2);write(char(x));
gotoxy(52,2);write(char(x));
gotoxy(50,5);write(char(x));
gotoxy(51,5);write(char(p));
gotoxy(50,5);write(char(p));
gotoxy(50,9);write(char(p));
```

```
gotoxy(51,9);write(char(p));
gotoxy(52,9);write(char(l));
    TEXTcolor(x);
gotoxy(54,3);write(char(p));   {A}
gotoxy(54,4);write(char(p));
gotoxy(54,5);write(char(p));
gotoxy(54,6);write(char(p));
gotoxy(54,7);write(char(x));
gotoxy(54,8);write(char(p));
gotoxy(54,9);write(char(p));
gotoxy(55,2);write(char(l));
gotoxy(56,2);write(char(l));
gotoxy(57,3);write(char(e));
gotoxy(57,4);write(char(p));
gotoxy(57,5);write(char(r));
gotoxy(57,6);write(char(d));
gotoxy(57,7);write(char(c));
gotoxy(57,8);write(char(x));
gotoxy(57,9);write(char(e));
gotoxy(55,5);write(char(l));
gotoxy(56,5);write(char(p));

gotoxy(59,2);write(char(s));   {N}
gotoxy(59,3);write(char(l));
gotoxy(59,4);write(char(r));
gotoxy(59,5);write(char(e));
gotoxy(59,6);write(char(e));
gotoxy(59,7);write(char(d));
gotoxy(59,8);write(char(c));
gotoxy(59,9);write(char(s));
gotoxy(60,3);write(char(l));
gotoxy(61,4);write(char(p));
gotoxy(62,2);write(char(p));
gotoxy(62,3);write(char(p));
gotoxy(62,4);write(char(p));
gotoxy(62,5);write(char(p));
gotoxy(62,6);write(char(p));
gotoxy(62,7);write(char(s));
gotoxy(62,8);write(char(x));
```

```
gotoxy(62,9);write(char(p));

gotoxy(12,13);write(char(s));   {I}
gotoxy(12,14);write(char(e));
gotoxy(12,15);write(char(r));
gotoxy(12,16);write(char(d));
gotoxy(12,17);write(char(c));
gotoxy(12,18);write(char(p));
gotoxy(12,19);write(char(d));
gotoxy(12,20);write(char(x));
gotoxy(12,11);write(char(p));

gotoxy(14,14);write(char(p));   {O}
gotoxy(14,15);write(char(d));
gotoxy(14,16);write(char(p));
gotoxy(14,17);write(char(s));
gotoxy(14,18);write(char(p));
gotoxy(14,19);write(char(p));
gotoxy(17,14);write(char(p));
gotoxy(17,15);write(char(p));
gotoxy(17,16);write(char(d));
gotoxy(17,17);write(char(p));
gotoxy(17,18);write(char(r));
gotoxy(17,19);write(char(e));
gotoxy(15,13);write(char(d));
gotoxy(16,13);write(char(x));
gotoxy(15,20);write(char(s));
gotoxy(16,20);write(char(l));

gotoxy(19,14);write(char(p));   {A}
gotoxy(19,15);write(char(p));
gotoxy(19,16);write(char(r));
gotoxy(19,17);write(char(e));
gotoxy(19,18);write(char(d));
gotoxy(19,19);write(char(c));
gotoxy(19,20);write(char(l));
gotoxy(22,14);write(char(s));
gotoxy(22,15);write(char(r));
gotoxy(22,16);write(char(c));
```

```
gotoxy(22,17);write(char(p));
gotoxy(22,18);write(char(p));
gotoxy(22,19);write(char(p));
gotoxy(22,20);write(char(p));
gotoxy(20,16);write(char(p));
gotoxy(21,16);write(char(x));
gotoxy(20,13);write(char(p));
gotoxy(21,13);write(char(p));

gotoxy(24,13);write(char(p));   {N}
gotoxy(24,14);write(char(p));
gotoxy(24,15);write(char(r));
gotoxy(24,16);write(char(e));
gotoxy(24,17);write(char(d));
gotoxy(24,18);write(char(c));
gotoxy(24,19);write(char(s));
gotoxy(24,20);write(char(l));
gotoxy(27,13);write(char(r));
gotoxy(27,14);write(char(x));
gotoxy(27,15);write(char(l));
gotoxy(27,16);write(char(p));
gotoxy(27,17);write(char(d));
gotoxy(27,18);write(char(p));
gotoxy(27,19);write(char(p));
gotoxy(27,20);write(char(p));
gotoxy(25,14);write(char(x));
gotoxy(26,15);write(char(p));

gotoxy(30,14);write(char(22));   {love-symbol}
gotoxy(30,15);write(char(2));
gotoxy(30,16);write(char(p));
gotoxy(31,13);write(char(p));
gotoxy(31,17);write(char(5));
gotoxy(32,12);write(char(6));
gotoxy(32,18);write(char(66));
gotoxy(33,12);write(char(63));
gotoxy(33,19);write(char(32));
gotoxy(34,13);write(char(2));
gotoxy(34,20);write(char(0));
```

```
gotoxy(35,19);write(char(p));
gotoxy(35,12);write(char(32));
gotoxy(36,18);write(char(45));
gotoxy(36,12);write(char(65));
gotoxy(37,13);write(char(52));
gotoxy(37,17);write(char(12));
gotoxy(38,14);write(char(2));
gotoxy(38,15);write(char(0));
gotoxy(38,16);write(char(p));

gotoxy(40,15);write(char(56));   {peace-symbol}
gotoxy(40,16);write(char(5));
gotoxy(40,17);write(char(222));
gotoxy(41,13);write(char(32));
gotoxy(41,19);write(char(44));
gotoxy(40,14);write(char(p));
gotoxy(40,18);write(char(p));
gotoxy(42,19);write(char(3));
gotoxy(42,12);write(char(p));
gotoxy(46,20);write(char(p));
gotoxy(46,12);write(char(3));
gotoxy(47,13);write(char(p));
gotoxy(47,19);write(char(2));
gotoxy(48,14);write(char(p));
gotoxy(48,18);write(char(p));
gotoxy(42,20);write(char(7));
gotoxy(43,12);write(char(77));
gotoxy(43,20);write(char(8));
gotoxy(44,12);write(char(3));
gotoxy(44,20);write(char(p));
gotoxy(45,12);write(char(p));
gotoxy(45,20);write(char(98));
gotoxy(46,19);write(char(65));
gotoxy(47,18);write(char(p));
gotoxy(48,15);write(char(111));
gotoxy(48,16);write(char(22));
gotoxy(48,17);write(char(p));

TEXTcolor(s);
```

```
gotoxy(44,12);write(char(4));
gotoxy(44,13);write(char(34));
gotoxy(44,14);write(char(p));
gotoxy(44,15);write(char(s));
gotoxy(44,16);write(char(4));
gotoxy(44,17);write(char(d));
gotoxy(43,18);write(char(e));
gotoxy(45,18);write(char(r));

gotoxy(54,12);write(char(s));   {cruce,cros}
gotoxy(54,13);write(char(l));
gotoxy(54,14);write(char(r));
gotoxy(54,15);write(char(d));
gotoxy(54,16);write(char(c));
gotoxy(54,17);write(char(p));
gotoxy(54,18);write(char(p));
gotoxy(54,19);write(char(32));
gotoxy(54,20);write(char(123));
gotoxy(53,20);write(char(38));
gotoxy(55,20);write(char(d));
gotoxy(50,13);write(char(e));
gotoxy(51,13);write(char(l));
gotoxy(52,13);write(char(s));
gotoxy(53,13);write(char(c));
gotoxy(55,13);write(char(p));
gotoxy(56,13);write(char(84));
gotoxy(57,13);write(char(p));
gotoxy(58,13);write(char(p));
gotoxy(54,11);write(char(22));
{gotoxy(54,10);write(char(p));}
gotoxy(45,20);write(char(33));
gotoxy(46,19);write(char(p));    gotoxy(58,17);write('by paul ioan muntean');
gotoxy(47,18);write(char(p));    gotoxy(58,18);write('2005 code_masters&CO
');
gotoxy(48,15);write(char(67));
gotoxy(48,16);write(char(45));
gotoxy(48,17);write(char(p));
gotoxy(43,20);write(char(188));
gotoxy(42,20);write(char(144));
```

```
until keypressed;
     clrscr;textbackground(blue); clrscr;
  gotoxy(23,14);
  write('what are you thinking now?');delay(3000);
   delay(1100);clrscr;
     textbackground(blue+181);
 for i:= 1 to 25 do
  begin
gotoxy(i,i);write('$ PM $');
  end;p:=0;
   for i:= 25 downto 1 do
      begin
      inc(p);
gotoxy(i,p);write('$ PM $');
      end;
gotoxy(23,12);write(' /////----------------------------------------\\\\\');
gotoxy(23,13);write('~~~~created by the smartest boy alias MASTERCODS
PM ~~~~');
gotoxy(23,14);write(' \\\\\----------------------------------------/////');
gotoxy(30,16);write('"PaulMuntean@gmail.com"-email addres ');
 gotoxy(35,17);write('creata in 6.5.2005');
delay(5000);gettime(h2,m2,s2,s1002);clrscr;  textbackground(blue); clrscr;
textcolor(yellow);
 end;
```

Problema 4,Meniul grafic.

Codul sursă:

```
Simulare asemanatoare cu cea din filmul Matrix;
 procedure gr3;
begin
   gettime(h1,m1,s1,s1001);
  textbackground(black);clrscr;
gotoxy(20,12);
writeln('ecranul va fi inundat!!!'); delay(3000);
clrscr;
textcolor(green);
repeat
x:=random(80);y:=random(80);
gotoxy(x,y);write('1');
```

```
z:=random(80);w:=random(80);
gotoxy(z,w);write('0');
until keypressed;clrscr;textcolor(magenta);
begin
  gotoxy(16,12);write('what is your sugestion?');delay(3000);clrscr;
for i:= 1 to 25 do
   begin
gotoxy(i,i);write('$ PM $');
   end;p:=0;
    for i:= 25 downto 1 do
       begin
       inc(p);
gotoxy(i,p);write('$ PM $');
       end;
 gotoxy(23,12);write(' /////----------------------------------------\\\\\');
gotoxy(23,13);write('~~~~created by the smartest boy alias MASTERCODS
PM ~~~~');
gotoxy(23,14);write(' \\\\\\----------------------------------------/////');
gotoxy(30,16);write('PaulMuntean@gmail.com');
 gotoxy(32,17);write('creata in 6.5.2005');
     delay(5000);
 gettime(h2,m2,s2,s1002);clrscr;
 end; gettime(h2,m2,s2,s1002); clrscr;   textbackground(blue); clrscr;
textcolor(yellow);
    end;
```

Problema 5,Meniul grafic.

Codul sursă:

```
Joc de numere sub forma unui triunghi;
   procedure gr4;
    begin
    n:=0;clrscr; write('APASA ENTER'); READLN; sound(1000);
 repeat
sound(n*100);
 textbackground(yellow);clrscr;
 textcolor(red);
 inc(n);
{ write('Dati valoarea lui n:');}{readln(N);}clrscr;
 begin
```

```
for i:=1 to n do
 begin
 for j:=1 to i do
  write(j:2);writeln;
 end;

 end;
begin
for i:=1 to n do
 begin
 for j:=1 to (n-i) do
  write(j:2);writeln
  end;
 end;

begin
for i:=1 to n do
 begin
 for j:=1 to i do
 gotoxy (n+n+n+1-j+1,j); write(j,' ');writeln;
 end;
 end;

begin
for i:=1 to n do
 begin
 for j:=1 to (n-i) do
  gotoxy (n+n+i+2,n+i);  write(j,' ');writeln
  end;
 end;
readln;
 until n=12;nosound;
   begin
 clrscr; gotoxy(16,12);write('what is your sugestion?');delay(3000);clrscr;
 for i:= 1 to 25 do
   begin
gotoxy(i,i);write('$ PM $');
   end;p:=0;
    for i:= 25 downto 1 do
```

```
      begin
      inc(p);
gotoxy(i,p);write('$ PM $');
      end;
  gotoxy(23,12);write(' /////-----------------------------------------\\\\\');
gotoxy(23,13);write('~~~~created by the smartest boy alias MASTERCODS
PM ~~~~');
gotoxy(23,14);write(' \\\\\-----------------------------------------/////');
gotoxy(30,16);write('PaulMuntean@gmail.com');
gotoxy(32,17);write('creata in 6.5.2005');
    delay(5000); textbackground(blue); clrscr;
    end;textbackground(blue); clrscr;textcolor(yellow);
    end;
```

Problema 1, Meniul cu altgoritmi matematici

Codul sursă:

```
Rezolva ecuatia de gr2 in C
procedure mat1;
 begin
clrscr;
 Write('Dati a: ');
 Readln(a);
 Write('Dati b: ');
 Readln(b);
 Write('Dati c: ');
 Readln(c);clrscr;
 If (a=0) then Writeln('ecuatia e de grI x1=',(-c/b):2)
 else
  Begin
  delta:=b*b-4*a*c;
    If delta<0 then
      begin
    writeln('x1=',(-
b/(2*a)):1,'+i',(sqrt(abs(delta))/(2*a)):1);
    writeln('x2=',(-b/(2*a)):1,'-
i',(sqrt(abs(delta))/(2*a)):1);
```

```
    delay(2000);
      end
     else
      If delta=0 then
        Begin
       x1:=-b/(2*a);
       Writeln('Solutia este: x1=',x1:8:4);
         Write('Solutia este: x2=',x1:8:4);
         End
                 else
                    Begin
                      x1:=(-b-sqrt(delta))/(2*a);
                      x2:=(-b+sqrt(delta))/(2*a);
                      Writeln('Prima solutie este x1=
',x1:8:4);
                      Writeln('A doua solutie este x2=
',x2:8:4);
                     delay(5000);
                     End; readln;
gettime(h2,m2,s2,s1002);
        End;
        end;
```

Problema 2, Meniul cu altgoritmi matematici

Codul sursă:

```
Bordarea unei matrice cu o linie si o coloana matrice

 procedure mat2;Inmultirea a 2 matrice
       begin
        clrscr;
S:=0;P:=0;L:=0;
writeln ('dati nr de linii ');readln(m);
writeln ('dati nr de coloane');readln(n);
for i:=1 to m do
 for j:=1 to n do
  begin
  write('dati elementul de pe linia ',i,' si coloana
',j,' =');readln(mat[i,j]);
  end;  clrscr;
 begin
```

```
 writeLN('MATRICEA INITIALA');
for i:=1 to m do
 begin
for j:=1 to n do
  write (mat[i,j]:3);
  writeln;
 end;
end;
     for i:=1 to m do
       begin          { S:=0;sau aici pus tot aia ii}
     for j:=1 to m do
         S:=S+mat[j,i];
         mat[m+1,i]:=S;S:=0;
       end;readln;
    for i:=1 to m do
     begin
     for j:=1 to n do
         P:=P+mat[i,j];
         mat[i,n+1]:=P;P:=0;
     end;
 n:=n+1;m:=m+1;ab:=0;
for i:=1 to m-1 do
begin
ab:=ab+mat[i,n-1];
end;
mat[m,n-1]:=ab;
for i:=1 to n-1 do
begin
L:=L+mat[m,i];
end;
mat[m,n]:=L;
readln;
 begin
 WRITELN('MATRICEA DUPA BORDAREA CU a ',M,' LINIE &
',N,'-a COLOANA');
for i:=1 to m do
 begin
for j:=1 to n do
begin
```

```
 if (i=m) or(j=n) then textcolor(green) ;
 write(mat[i,j]:3);
 end;  writeln  ;textcolor(yellow);
end;
end;gettime(h2,m2,s2,s1002);
readln;
       end;
```

Problema 3, Meniul cu altgoritmi matematici

Codul sursă:

```
Transformarea unui nr in orice baza

   procedure mat3;
  begin
         gettime(h1,m1,s1,s1001);clrscr;
      write('Dati baza dorita!');readln(b);
      writeln('Dati nr. pe care doriti sa-l transformati
in baza: ',b);readln(ns);
        i:=1;
      while ns>0 do
         begin
         vl[i]:=(ns mod b);
         inc(i);
         ns:=ns div b;
         end;
       n2:=i-1;
            for i:=n2 downto 1 do
        begin
       write(vl[i]);
        end;readln;gettime(h2,m2,s2,s1002);
       end;
```

Problema 4, Meniul cu altgoritmi matematici

Codul sursă:

```
  procedure mat4;
       begin
           {
  Programul realizeaza inmultirea a doua matrici ce are
elemente nr intregi.
```

```
  Programul avertizeaza asupra compatibilitati
matricelor ce trebuiesc inmulti
  te si afiseaza fiecare matrice.  }
      gettime(h1,m1,s1,s1001); clrscr;
  repeat
  writeln('  Programul realizeaza inmultirea a doua
matrici ce are elemente nr intregi.');
  writeln('  Programul avertizeaza asupra
compatibilitati matricelor ce trebuiesc inmulti');
  writeln('  te si afiseaza fiecare
matrice.');delay(10000);
   write('Nr de linii a matricei A: ');readln(m);
   write('Nr de coloane a matricei A: ');readln(n);
   write('Nr de linii a matricei B: ');readln(k);
   write('Nr de coloane a matricei B: ');readln(p);
   if k<>n then writeln('Atentie nr de coloane a lui A =
nr linii a lui B !');
  until n=k;
     for i:= 1 to m do
      for j:= 1 to n do
        begin
        write('A1[',i,',',j,']= ');
        readln(A1[i,j]);
        end;
   write('Matricea A este : ');
   begin
   for i:= 1 to m do
     begin
     writeln;
      for j:= 1 to n do
       write(A1[i,j],' ');
     end;
      writeln;
    end;
    for i:= 1 to k do
      for j:= 1 to p do
        begin
        write('B1[',i,',',j,']= ');
        readln(B1[i,j]);
```

```
     end;
write('Matricea B este : ');
begin
for i:= 1 to k do
begin
writeln;
for j:= 1 to p do
    write(B1[i,j],' ');
  end;
   writeln;
end;
     begin
 for i:= 1 to n do
  for j:= 1 to k do
    C1[i,j]:= 0
     end;
  for s:=1 to m do
   for w:=1 to p do
     for j:= 1 to n do
       C1[s,w]:= C1[s,w]+(A1[s,j]*B1[j,w]);
write('MATRICEA CE REZULTA ESTE : ');
         begin
for i:= 1 to m do
  begin
  writeln;
   for j:= 1 to p do
    write(C1[i,j],'   ');
  end;
   writeln;
 end;
readln;gettime(h2,m2,s2,s1002);
     end;
```

Problema 5, Meniul cu altgoritmi matematici

Codul sursă:

```
Implementeaza functia Ackerman

procedure functia_Ackerman;
```

```
          {Calculul functiei recursive Aackerman
definita prin relatiile:}
{Ack(m,n)=     n+1 , daca m=0 }
{              Ack(m-1,1) ,daca m>0 si n=0    }
{              Ack(m-1, ack(m,n-1)) daca n>0, m>0}
{Este un exemplu de functie recursiva implementat prin
Function ac}
{Program functia_Ackerman; }
function Ac(m,n:longint):longint;
begin
if m=0 then ac:=n+1
       else if n=0 then ac:=ac(m-1,1)
                   else ac:=ac(m-1,ac(m,n-1));
end;
begin
clrscr;
writeln('Calculul functiei recursive Aackerman definita
prin relatiile:');
writeln('Ack(m,n)=     n+1 , daca m=0 ');
writeln(' Ack(m-1,1) ,daca m>0 si n=0');
writeln('  Ack(m-1, ack(m,n-1)) daca n>0, m>0' );
textcolor(yellow);
textbackground(red);
write('a= ');readln(a);
write('b= ');readln(b);
writeln('Ackerman din ',a,' si ',b,'=');
write( Ac(a,b));delay(2000);
readln;
readln;
end;
```

Problema 6, Meniul cu altgoritmi matematici

Codul sursă:

```
Implementeaza functia Mana_Pneulli
procedure Mana_Pnuelli;
function pnuelli(x:integer):integer;
begin
if x<12  then pnuelli:=pnuelli(pnuelli(x+2))
       else pnuelli:=x-1;
```

```
end;
begin
clrscr;
write('n= ');read(n);
write('Rezultatul este: ', pnuelli(n));readln;
readln;
end;
```

Problema 1.Meniul cu fisiere

Codul sursă:

```
Pprogram pt aranjarea alfabetice a unei liste dintr-un fisier text
PROCEDURE  Ara;

   procedure citirea;
   var p:byte;
    begin
    p:=1;
    while not eof(f) do
      begin
    readln(f,v[p]);inc(p);
      end;n:=p;
    end;
   procedure aranjare_afisare;
    begin
   for i:=1 to n-1 do {ordonarea vectorului}
    for j:=1 to n do
      if v[j]<v[i] then
        begin{schimbarea}
        aux:=v[i];
        v[i]:=v[j];
        v[j]:=aux;
        end;textcolor(magenta);
        writeln('Ordinea alfabetica a listei cu',n-2,'elemente
este:');textcolor(green);  j:=0;
       for i:=n-2 downto 1 do
          begin
             inc(j);
```

```
        write(j:3);
       writeln(v[i]);
       end;

end;
procedure completareFIS;
 begin
 writeln(g,'Ordinea alfabetica a listei compusa din-',n-2,'-elemente este:');
     j:=0;
 for i:=n-2 downto 1 do
  begin
    inc(j);
 writeln(g,j,':',v[i]);
  end;
 end;
   begin
   clrscr;
   assign(f,'date.DOC');reset(f);
   assign(g,'rezultat.DOC');rewrite(g);
   citirea;write('Se prelucreaza');
   aranjare_afisare;
   completareFIS;
   close(f);
   close(g);
   end;
```

Problema 2.Meniul cu fisiere

Codul sursă:

Simularea click 2 de la mouse

```
procedure click2;
```

```
PROCEDURE MOUSE;
  begin
assign(f,nume);rewrite(f);
   end;
    BEGIN
    write('SE SIMULEAZA FUNCTIONAREA CLICK 2 DE LA MOUSE SI
APOI NEW');
WRITEln('SE POATE CREA IN DIRECRORUL RADACINA
URMATOARELE TIPURI DE FISIERE');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,JPEG
EX:GEORGE.JPEG');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,BMP
EX:GEORGE.BMP');
WRITEln('Microsoft Excel Worksheet,TASTATI NUMELE FISIER SI
EXTENSIA,XLS EX:GEORGE.XLS');
WRITEln('Microsoft Office Acces ,TASTATI NUMELE FISIER SI
EXTENSIA,MDB EX:GEORGE.MDB');
WRITEln('Microsoft Excel Worksheet,TASTATI NUMELE FISIER SI
EXTENSIA,XLS EX:GEORGE.XLS');
WRITEln('Microsoft Office Publisher,TASTATI NUMELE FISIER SI
EXTENSIA,PUB EX:GEORGE.PUB');
WRITEln('Microsoft Word Document,TASTATI NUMELE FISIER SI
EXTENSIA,DOC EX:GEORGE.doc');
WRITEln('Microsoft PowerPoint ,TASTATI NUMELE FISIER SI
EXTENSIA,PPT EX:GEORGE.PPT');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,TXT
EX:GEORGE.TXT');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,WAV
EX:GEORGE.WAV');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,RAR
EX:GEORGE.RAR');
WRITEln('TASTATI NUMELE FISIER SI EXTENSIA,ZIP
EX:GEORGE.ZIP');textcolor(yellow);
writeln('Dati numele dorit si extensia');readln(nume);
  MOUSE;write('S-a creat,doriti sa-l stergeti
[Y/N]?');readln(tas);textcolor(yellow);
  if (tas=chr(121)) then erase(f); textcolor(yellow);
   END;
```

Problema 3.Meniul cu fisiere

Codul sursă:

```
Generarea unei parole
procedure genParola;
 begin

 v1[1].carac:='A';
 v1[2].carac:='B';
 v1[3].carac:='C';
 v1[4].carac:='D';
 v1[5].carac:='E';
 v1[6].carac:='F';
 v1[7].carac:='G';
 v1[8].carac:='H';
 v1[9].carac:='I';
 v1[10].carac:='J';
 v1[11].carac:='K';
 v1[12].carac:='L';
 v1[13].carac:='M';
 v1[14].carac:='N';
 v1[15].carac:='O';
 v1[16].carac:='P';
 v1[17].carac:='Q';
 v1[18].carac:='R';
 v1[19].carac:='S';
 v1[20].carac:='T';
 v1[21].carac:='U';
 v1[22].carac:='V';
 v1[23].carac:='W';
 v1[24].carac:='X';
 v1[25].carac:='Y';
```

```
 v1[26].carac:='Z';
 v1[27].carac:='0';
 v1[28].carac:='1';
 v1[29].carac:='2';
 v1[30].carac:='3';
 v1[31].carac:='4';
 v1[32].carac:='5';
 v1[33].carac:='6';
 v1[35].carac:='7';
 v1[34].carac:='8';
 v1[36].carac:='9';
 v1[37].carac:='0';
 j:=0 ;
 assign(f,'parola.doc');rewrite(f);
 writeln('Se genereaza parola tip jocuri ex:FIFA');
randomize;
  REPEAT
  i:=random(37);
   inc(j); i:=random(37);
  write(f,v1[i].carac);
        i:=random(37); i:=random(37);
   write(v1[i].carac);if j mod 4=0 then begin write('
');write(f,' ');end;  i:=random(37);
 until j=20;
 close(f);writeln(' ');writeln('Parola a fost copiata in
Parola.doc din directorul radacina');readln;
end;
```

Problema 4.Meniul cu fisiere

Codul sursă:

```
Unplerea aleatoare a unui fisier
procedure umplere_fisier_text;
begin
 v1[1].carac:='A';
 v1[2].carac:='B';
 v1[3].carac:='C';
 v1[4].carac:='D';
```

```
v1[5].carac:='E';
v1[6].carac:='F';
v1[7].carac:='G';
v1[8].carac:='H';
v1[9].carac:='I';
v1[10].carac:='J';
v1[11].carac:='K';
v1[12].carac:='L';
v1[13].carac:='M';
v1[14].carac:='N';
v1[15].carac:='O';
v1[16].carac:='P';
v1[17].carac:='Q';
v1[18].carac:='R';
v1[19].carac:='S';
v1[20].carac:='T';
v1[21].carac:='U';
v1[22].carac:='V';
v1[23].carac:='W';
v1[24].carac:='X';
v1[25].carac:='Y';
v1[26].carac:='Z';
v1[27].carac:='0';
v1[28].carac:='1';
v1[29].carac:='2';
v1[30].carac:='3';
v1[31].carac:='4';
v1[32].carac:='5';
v1[33].carac:='6';
v1[35].carac:='7';
v1[34].carac:='8';
v1[36].carac:='9';
v1[37].carac:='.';
v1[38].carac:=',';
v1[39].carac:=':';
v1[40].carac:='?';
v1[41].carac:=chr(41);
v1[42].carac:='-';
v1[43].carac:='/';
```

```
 v1[44].carac:=chr(40);
 v1[45].carac:=chr(41);
 v1[46].carac:=' ';
 assign(f,'Umplut.txt');rewrite(f); randomize;
 write(f,'Fisieul  a fost umplut');
 writeln('Fisieul  se va umple pana se apasa
ENTER');readln;
 repeat
 i:=random(46);
 write( v1[i].carac);
 write( f,v1[i].carac);
 until keypressed;
 clrscr;readln;
 write('Fisieul  a fost umplut');readln;
  close(f);
  end;
```

Problema 5.Meniul cu fisiere

Codul sursă:

Generarea unei imagini cu Arnold Schwarzeneger

```
   procedure arnold;
 var f:text;
Procedure arn;
begin
 assign(f,'text.txt');rename(f,'ARNOLD.htm');
 WRITEln('APASA ALT+TAB  SI UITATE IN FISIERUL CU PROGRAMUL
LA "ARNOLD.HTM"'); DELAY(25000);rename(f,'text.txt');
   write('SA REZOLVAT');
  end;
   begin
  arn;READLN;
   end;
```

Problema 1.Meniul cu grafuri

Codul sursă:

```
Verifica daca un graf este sau nu hamitonian
procedure graf_hamiltonian;
procedure citeste;
begin
for i:=1 to n do
for j:=1 to n do begin
al[i,j]:=0;al[i,i]:=1;end;
for i:=1 to m do
 begin
 write('dati extremitatile muchiei',i);
   read(x,y);
   al[x,y]:=1;
   al[y,x]:=1;
 end;
end;
function grad(x:integer):integer;
var i,s:integer;
begin
s:=0;
for i:=1 to n do
if al[x,i]=1 then s:=s+1;
grad:=s;
end;
begin
writeln('Dati nr de linii si coloane');readln(n);
writeln('Dati nr de muchii');readln(m);
citeste;
ok:=true;
for i:=1 to n do
begin
if grad(i)<n div 2 then ok:=false;
if ok=false then writeln('Graful nu este hamiltonian')
          else writeln('Graful este hamiltonian');
end;readln;
```

```
end;
```

Problema 2.Meniul cu grafuri

Codul sursă:

```
Afisarea tuturor componentelor tare conexe dintr-un graf ;
 procedure componenta_tareconex;
Procedure citire;
begin
for i:=1 to n do
for j:=1 to n do begin
al[i,j]:=0;al[i,i]:=1;end;
for i:=1 to m do
 begin
 write('dati extremitatile muchiei',i);
   read(x,y);
   al[x,y]:=1;
   al[y,x]:=1;
 end;
for i:=1 to n do
    xi[i]:=i;
end;

begin
writeln('Dati nr de linii si coloane');readln(n);
writeln('Dati nr de muchii');readln(m);
citire;
for i:=1 to m do
   if xi[ap[i].i]<xi[ap[i].j] then begin
                          for j:=1 to n do
                             if xi[j]=xi[ap[i].j] then
                                          xi[j]:=xi[ap[i].i];
                          end
   else if xi[ap[i].i]>xi[ap[i].j] then begin
                          for j:=1 to n do
                             if xi[j]=xi[ap[i].i] then
                                          xi[j]:=xi[ap[i].j];
                          end;
j:=1;
```

```
for i:=1 to n-1 do
   if xi[i]<>xi[i+1] then j:=j+1;
writeln('Acest graf are ',j,' componente conexe');readln;
for i:=1 to n do
   write(xi[i],' ');
writeln;
writeln;
readln;
end;
```

Problema 3.Meniul cu grafuri

Codul sursă:

```
Afisarea subgrafurilor conexe din cel initial
procedure subGrafConex;
 Procedure citire;
begin
for i:=1 to n do
for j:=1 to n do begin
al[i,j]:=0;al[i,i]:=1;end;
for i:=1 to m do
  begin
 writeln('Dati extremitatile muchiei ',i);
   read(x,y);
   al[x,y]:=1;
   al[y,x]:=1;
  end;
 end;
 procedure subgrafconex;
 var i,p,q,start:integer;
 pus:array[1..20] of boolean;
  begin
  for i:=1 to n do pus[i]:=false;
  write('Varful de plecare este:');readln(start);p:=start;pus[p]:=true;
    repeat
     q:=0;
       repeat
       q:=p+1;
       until (q>n) or ((pus[q]=false )and(al[p,q]=1));
```

```
      if (not pus[q]) and (al[p,q]=1 )then
       begin
       writeln(q);pus[q]:=true;p:=q;
       end;
  until q>n;textcolor(red);
  Writeln('Varful care poate fi eliminat pt. ca graful sa fie conex
este:',p);readln;
  end;
 begin
writeln('Dati nr de linii si coloane');readln(n);
writeln('Dati nr de muchii');readln(m);
citire;writeln; subgrafconex;
 end;
```

Problema 4.Meniul cu grafuri

Codul sursă:

```
Verifica daca un graf este eulerian
procedure grafEulerian;
  Procedure citire;
begin
for i:=1 to n do dv[i]:=0;
for i:=1 to n do
for j:=1 to n do begin
al[i,j]:=0;al[i,i]:=1;end;
for i:=1 to m do
  begin
 writeln('Dati extremitatile muchiei ',i);
   readln(x,y);
   al[x,y]:=1;
   al[y,x]:=1;
   dv[x]:=dv[x]+1;dv[y]:=dv[y]+1;
  end;
 end;
 function nevizitat:Integer;
 var j,pi,ps,p,nc:integer;
 begin
 prim_nev:=-1;
 j:=1;
 while(j<=n)and(prim_nev=-1) do
```

```
  begin
  if (viz[j]=0)then
  prim_nev:=j;
  inc(j);
  end;
  nevizitat:=prim_nev;
 end;
 function conex:boolean;
 var k,pi,ps,z:integer;
 begin
 for k:=1 to 20 do Cv[k]:=0;
  for p:=1 to 20 do viz[p]:=0;
  write('Varful de plecare este:');readln(prim);
  pi:=1;ps:=1;
  Cv[1]:=prim;viz[prim]:=1;
  while ps<=pi do
    begin
    z:=Cv[ps];
    for k:=1 to n do
    if (al[z,k]=1)and(viz[k]=0)then
      begin
      inc(pi);Cv[pi]:=k;viz[k]:=1;
      end;
      inc(ps);
     end;
    for k:=1 to pi do write(Cv[k]:3);writeln;
    if nevizitat=-1 then conex:=true
                    else conex:=false;
 end;
 function grade_pare:boolean;
 var i:integer;ok:boolean;
 begin
 ok:=true;
 for i:=1 to n do
 if dv[i] mod 2<>0 then ok:=false;
     grade_pare:=ok;
 end;
begin
writeln('Dati nr de linii si coloane');readln(n);
```

```
writeln('Dati nr de muchii');readln(m);
citire;writeln;
 writeln('Afisam componentele conexe');
 if conex=true then
 if grade_pare=true then writeln('Graful este conex si eulerian')
                else writeln('Graful este conex dar nu si eulerian')
                else Writeln('Graful nu este conex');readln;
 end;
```

Problema 5.Meniul cu grafuri

Codul sursă:

```
Afisarea arborelui partial de cost minim
procedure arb_partial_de_cost_minim;
type vec=array[1..30] of integer;
var a:array[1..30,1..30] of integer;
s,t,c:vec;n:integer;
procedure citire_matrice;
var i,j:integer;
begin
write('Dati nr de vf:');readln(n);
for i:=1 to n do a[i,i]:=0;
for i:=1 to n do
for j:=i+1 to n do
  begin
  write('Cost a[',i,',',j,']=');readln(a[i,j]);
a[j,i]:=a[i,j];
  end;
end;
procedure afisare_matrice;
var i,j:integer;
begin
writeln('Graful are',n,'varfuri');
writeln('Matricea costurilor este:');
 for i:= 1 to n do
     begin
      for j:= 1 to n do
       write(a[i,j],'   ');
       writeln;
    end;
```

```
end;
procedure afisare_arbore(mesaj:string;v:vec;n:integer);
var i:integer;
begin
writeln(mesaj);
for i:=1 to n do write(v[i]:2);
writeln;
end;
procedure formare_arbore;
var k,i,j,start,cost_min,n1,n2:integer;
begin
  for i:=1 to n do
   begin
s[i]:=0;t[i]:=0;c[i]:=0;
   end;
write('Dati varful de start:');readln(start);s[start]:=1;
for k:=1 to n-1 do
   begin
 cost_min:=maxint;n1:=-1;n2:=-1;
  for i:= 1 to n do
  for j:= 1 to n do
  if (s[i]=0 )and (s[j]=0) then
  if a[i,j]<>0 then
  if a[i,j]<cost_min then
     begin
  cost_min:=a[i,j];n1:=i;n2:=j;
     end;
  s[n2]:=1;t[n2]:=n1;c[n2]:=a[n1,n2];
  end;
end;
   begin
citire_matrice;afisare_matrice;formare_arbore;
afisare_arbore('Vectorul caracteristic  S este ',s,n);
afisare_arbore('Vectorul parintilor  T este ',t,n);
afisare_arbore('Vectorul costurilor C este ',c,n);readln
   end;
```

Problema 1.Meniul BKTR

Codul sursă:

```
Afisarea permutarilor de n elemente
procedure bktr_nerecursiv_permutari;
type vector=array[1..25] of integer;
var st, v:vector;
   n:integer;

procedure initializari;
var i:integer;
begin
    write('n='); readln(n);
    for i:=1 to 25 do st[i]:=0;
    writeln;
end;

procedure tipar(p:integer);
var i:integer;
begin
    for i:=1 to p do write(st[i]:4,' ');
    writeln;
end;

function valid(p:integer):boolean;
var i:integer; ok:boolean;
begin
    ok:=true;
    for i:=1 to p-1 do
       if st[p]=st[i] then ok:=false;
    valid:=ok;
end;

procedure back; {implementeaza algoritmul nerecursiv de backtracking}
var p:integer;  {varful stivei}
begin
    p:=1; st[p]:=0; {initializam primul nivel}
    while p>0 do    {cat timp stiva nu devine din nou vida}
     begin
       if st[p]<n then
         begin
            st[p]:=st[p]+1; {punem pe nivelul p urmatoarea valoare}
```

```
            if valid(p) then {daca solutia(st[1],st[2],...,st[p]) este valida}
              if p=n then  {daca solutia este si finala}
                tipar(p)
               else
                begin
                 p:=p+1; st[p]:=0;{trecem la nivelul urmatoor, pe care il
reinitializam}
                end;
         end
           else
        p:=p-1; {pasul inapoi la nivelul anterior}
     end;
end;

begin
    clrscr;
    initializari;
    back;
    readln;
end;
```

Problema 2.Meniul BKTR

Codul sursă:

```
Afisarea combinarilor de n elemente luate cate k
procedure bktr_combinari;
type vector=array[1..100] of integer;
var st:vector;
   n,k:integer;

procedure initializari;
var i:integer;
begin    repeat
    write('n='); readln(n);
     write('k='); readln(k);
     until n>=k;
    for i:=1 to 25 do st[i]:=0;
    writeln;
end;
```

```
procedure tipar(p:integer);
var i:integer;
begin
    for i:=1 to p do write(st[i]:4,' ');
    writeln;
end;

function valid(p:integer):boolean;
var i:integer; ok:boolean;
begin
    ok:=true;
    for i:=1 to p-1 do
       if st[p]=st[i] then ok:=false;
    valid:=ok;
end;

procedure bktr(p:integer); {implementeaza algoritmul nerecursiv de
backtracking}
var pval:integer;  {varful stivei}
begin
  for pval:=1 to n do
     begin
     st[p]:=pval;
     if valid(p) then
     if p=k then
     tipar(p)
     else bktr(p+1);
     end;
end;

begin
    clrscr;
    initializari;
    bktr(1);
    readln;
end;
```

Problema 3.Meniul BKTR

Codul sursă:

```
Afisarea produsului cartezian a n multimi
procedure bktr_prodCartezian;
type vector=array[1..100] of integer;
var st,nr:vector;
   n,k:integer;

procedure initializari;
var i:integer;
begin
    write('n='); readln(n);
     for i:=1 to 50 do st[i]:=0;
      for i:=1 to n do
      begin
      writeln('Nr de lemente a multimi ',i);readln(nr[i]);
      end;
end;

procedure tipar(p:integer);
var i:integer;
begin
    for i:=1 to p do write(st[i]:4,' ');
    writeln;
end;

function valid(p:integer):boolean;
var i:integer; ok:boolean;
begin
    valid:=true;
end;

procedure bktr(p:integer); {implementeaza algoritmul nerecursiv de
backtracking}
var pval:integer;  {varful stivei}
begin
 if p=n+1 then tipar(p-1)
        else
```

```
  for pval:=1 to nr[p] do
     begin
     st[p]:=pval;
     if valid(p) then
     bktr(p+1);
     end;
end;

begin
    clrscr;
    initializari;
    bktr(1);
    readln;
end;
```

Problema 4.Meniul BKTR

Codul sursă:

```
Generarea numerelor prime formate din numerele0,2,9
procedure bktr_Nrprime;
const u:array[1..3] of integer=(0,2,9);
type vector=array[1..100] of integer;
var st:vector;
   n:integer;

procedure initializari;
var i:integer;
begin
writeln('Se genereaza toate nr. prime alcatuite din 3 cifre cu nr. 0,2,9');
    writeln('Dati nr de cifre a numerelor n='); readln(n);
     for i:=1 to 50 do st[i]:=0;

end;
function test_prim(x:integer):boolean;
var k:integer;
begin
test_prim:=true;
for k:=2 to x div 2 do
if x mod k=0 then test_prim:=false;
end;
```

```
function putere(a,m:integer):integer;
var f,p:integer;
begin
P:=1;
for f:=1 to m do
 P:=P*a;
 putere:=P;
end;
procedure tipar(p:integer);
var j,y:integer;
begin
Y:=0;
    for j:=n downto 1 do y:=y+st[j]*putere(10,j-1);
    if (test_prim(y)=true) and (st[p]<>0) then writeln(y);

end;

function valid(p:integer):boolean;
var i:integer; ok:boolean;
begin
    valid:=true;
end;

procedure bktr(p:integer); {implementeaza algoritmul nerecursiv de
backtracking}
var pval:integer;  {varful stivei}
begin
  for pval:=1 to 3 do
     begin
     st[p]:=u[pval];
     if valid(p) then
     if p=n then
     tipar(p)
     else
     bktr(p+1);
     end;
end;

begin
```

```
    clrscr;
    initializari;
    bktr(1);
    readln;
end;
```

Problema 5.Meniul BKTR

Codul sursă:

```
Problema comis voiajorului
procedure bktr_comisVOIAJOR;
type vector=array[1..100] of integer;
matice=array[1..50,1..50] of 0..1;
var st:vector;
   n,start:integer;a:matrice;

procedure initializari;
var i,i0,j0:integer;
begin
writeln('Se considera n orase numerotate de la 1..n.Un comis voiajor trebuie');
writeln('sa-si prezinte produsele in toate cele n orase plecand dintr-un oras de
start');
writeln('tecand prin fiecare oras o data si revenind in orasul de start');
writeln('Sa se afiseze toate traseele  pe care le poate urma');
    write('Nr oraselor:'); readln(n);
   write('Orasul de start:'); readln(start);
     for i:=1 to 50 do st[i]:=0;
      for i0:=1 to n do a[i0,i0]:=0;
         for i0:=1 to n do
          for j0:=1 to n do
          if i0<j0 then
           begin
           write('a[',i0,',',j0,']=');readln(a[i0,j0]);
           a[j0,i0]:=a[i0,j0];
            end;
            st[1]:=start;
end;

procedure tipar(p:integer);
var i:integer;
```

```
begin
for i:=1 to p do write(st[i]:4,' ');
writeln;
end;

function valid(p:integer):boolean;
var i:integer; ok:boolean;
begin
    ok:=true;
    for i:=1 to p-1 do
    if st[p]=st[i] then ok:=false;
    if a[st[p],st[p-1]]=0 then ok:=false;
    valid:=ok;
end;

procedure bktr(p:integer); {implementeaza algoritmul nerecursiv de
backtracking}
var pval:integer;  {varful stivei}
begin
  for pval:=1 to n do
     begin
     st[p]:=pval;
     if valid(p) then
     if(p=n) and(a[st[p],start]=1) then
     tipar(p)
     else
     bktr(p+1);
     end;
end;

begin
    initializari;
    bktr(2);
    readln;
end;
```

Problema 1.Meniul Divide et Impera

Codul sursă:

Cautarea binara intru-n sir ordonat crescator sau descrescator

```
procedure cautare_binara;
type sir=array[1..10] of integer;
var a:sir;
   i,n:byte;
   x:integer;
procedure   cautare(st,dr:byte);
var mijloc:byte;
begin
if st>dr then  begin
write('Valoare ',x,' nu apartine sirului');exit;
end;
mijloc:=(st+dr) div 2;
if a[mijloc]=x then
write('valoarea ',x,' se afla pe pozitia  ',mijloc)
            else
if a[mijloc]>x then
cautare(st,mijloc-1)
                        else
cautare(mijloc+1,dr);
end;
begin
writeln('Dati nr de valori ');read(n);
writeln('Dati sirul ordonat:');
for i:=1 to n do
begin
write('a[',i,']= ');
read(a[i]);
end;
write('Dati valoarea cautata');read(x);
cautare(1,n);
readln;
readln;
end;
```

Problema 2.Meniul Divide et Impera

Codul sursă:

Problema turnurilor din hanoi

```
procedure turnurile_hanoi;
```

```
var x,y,z:char;n:integer;
procedure hanoi(n:integer;x,y,z:char);
 begin
  if n=1 then
  writeln(x,y)
   else
   begin
   hanoi(n-1,x,z,y);
   writeln(x,y);
   hanoi(n-1,z,y,x);
   end;
 end;
 begin
 writeln('Se dau 3 tije, simbolizate prin X,Y,Z se dau n discuri de diametre
diferite');
 writeln ('diametre diferite stivuite pe tija  X,in ordinea descrescatoare ');
 writeln ('a diametrelor,formand un turn.Sa se mute cele n discuri pe tija Y,');
 writeln('folosind tija intermediara z si respectand regulile:');
 writeln ('1:in fiecare miscare se muta 1 disc');
 writeln('2:un disc nu poate fi asezat pe unul cu diametru mai mic');
 write ('n=');readln(n);
 hanoi(n,'X','Y','Z');READLN;
 END;
```

Problema 3.Meniul Divide et Impera

Codul sursă:

Cel mai mare divizor comun a n numere

```
 procedure CmmdcDEnNR;
 var x:array[1..500] of integer;
 i,n:integer;
 function cmmdc(u,v:integer):integer;
 begin
 if u=v then cmmdc:=u  else
 if u>v then cmmdc:=cmmdc(u-v,v)
        else
        cmmdc:=cmmdc(u,v-u);
 end;
 function div_imp(p,q:integer):integer;
```

```
 var m,d1,d2:integer;
 begin
 if abs(p-q)<=1 then
 div_imp:=cmmdc(x[p],x[q])
   else
    begin
    m:=(p+q)div 2;
    d1:=div_imp(p,m);
   d2:=div_imp(1+m,q);
   div_imp:=cmmdc(d1,d2);
   end;
  end;
  begin
  write('Dati nr. de numere:');readln(n);
  Writeln('Se calculeaza cmmdc din ',n,'elemente  mai mari ca "0"');
   for i:=1 to n do
    repeat
    write('x[',i,']=');readln(x[i]);
    until x[i]>0;
    writeln('Cel mai mare divizor comun a numerelor
este:',div_imp(1,n));readln;
    end;
```

Problema 4.Meniul Divide et Impera

Codul sursă:

Program de aflare a coordonatelor unde trebuie sa fie taiata o tabla ca ea sa nu mai aiba nici o gaura

```
 Procedure foaiatabla;
 var x,y:array[1..200] of integer;
     n:integer;
  function max(i,j:integer):integer;
  begin
  if i<j then max:=j
        else max:=i;
  end;
function arie(x1,y1,x2,y2:integer):integer;
var i:integer;neel:boolean;
begin
i:=1; neel:=false;
```

```
repeat
 neel:=(x1<x[i])and(x[i]<x2)and(y1<y[i])and(y[i]<y2);inc(i);
until neel or (i=n+1);
dec(i);
if not neel then arie:=(x2-x1)*(y2-y1)
else
arie:=max(max(arie(x1,y1,x[i],y2),arie(x[i],y1,x2,y2)),max(arie(x1,y1,x2,y[i]),ari
e(x1,y[i],x2,y2)));
end;
begin
writeln('Se da o bucata dreptunghiulara de tabla cu lungime l si inaltime
h,avand ');
writeln('pe suprafata ei n gauri cu coodonatele lor.Sa se decupeze din tabla o
bucata ');
writeln('de arie maxima care sa nu aiba gauri sunt permise numai taieturi');
writeln('orizontale si verticale');textcolor(red);
write('Dati nr de gauri');READLN(N);textcolor(yellow);
for i:=1 to n do
begin
writeln('Coordonatele punctului: ',i);readln(x[i],y[i]);
end;clrscr;
writeln('Aria maxima este',arie(0,0,100,100));readln;
end;
```

Problema 5.Meniul Divide et Impera

Codul sursă:

Sortarea rapida a unui vector de numere

```
procedure quick_sort;
var v:array[1..100] of integer;
n,i:integer;
procedure quick(s,d:integer);
var a,b,t,ia:integer;
begin
a:=s;b:=d;
repeat
while v[a]<v[b] do dec(b);
t:=v[a];v[a]:=v[b];v[b]:=t;
inc(a);ia:=1;
 if a<b then
```

```
 begin
 while v[a]<v[b] do inc(a);
 t:=v[a];v[a]:=v[b];v[b]:=t;
 dec(b);ia:=0;
 end;
 until b<=a;
 if a<a-ia then quick(s,a-ia);
 if a-ia+1<d then quick(a-ia+1,d);
end;
begin
writeln('Sortarea unui sir prin metoda quick sort');
write('Dati nr. de elemente:');readln(n);
for i:=1 to n do
begin
write('v[',i,']=');readln(v[i]);
end;
quick(1,n);
for i:=1 to n do write(v[i]:2);
readln;
end;
```

Problema 1.Meniul cu altgoritmi speciali

Codul sursă:

Afisarea pe ecran a unei flacari formata di pixeli

```
  procedure flame;
{$ifndef __DOS__}
 { This program can be compiled for MS-DOS target only;}
{$endif}

{ this exaple demonstrate direct access to video memory }
{ and I/O port                                          }

{ this program has been posted to COMP.LANG.PASCAL newsgroup.}
{ Here is unchanged original version with author comments.   }
{ This program can't works under DOS32 extender             }

{ DOS version: Use Turbo Pascal 6.0+ to compile }

var c, x, y, z : Word;
```

```
procedure setrgb( c, r, g, b : byte );
begin

  port[$3c8] := c;   { g'day, this is a probably the most simple version   }
  port[$3c9] := r;   { of fire that you will ever see in pascal. i wrote   }
  port[$3c9] := g;   { the code in pascal so it's slow and choppy, i have  }
  port[$3c9] := b;   { another version in asm. and it's faster. anyways if }
end;               { you have any critics or question on this code, just }
                   { e-mail me at ekd0840@bosoleil.ci.umoncton.ca. or    }
begin              {            9323767@info.umoncton.ca              }
  randomize;       {  note : I have code for all kinds of stuff (that I  }
  asm   mov ax, 13h  {        wrote of course), if you want something     }
        int 10h      {        e-mail me (i never get mail), maybe i have  }
  end;             {        what you want.                              }
  for x := 1 to 32 do{                        keith degr□ce        }
  begin            {                        moncton, n.-b. canada }
    setrgb(x,   x*2-1, 0,     0    );
    setrgb(x+32, 63,   x*2-1, 0    );
    setrgb(x+64, 63,   63,    x*2-1);
    setrgb(x+96, 63,   63,    63   );
  end;
  port[$60] := 0;
  repeat
   x := 0;
   repeat
     y := 60;
     repeat
       c := (mem[$a000:y * 320 + x]+
            mem[$a000:y * 320 + x + 2]+
            mem[$a000:y * 320 + x - 2]+
            mem[$a000:(y+2) * 320 + x + 2]) div 4;
       if c <> 0 then dec(c);
       memw[$a000:(y-2) * 320 + x] := (c shl 8) + c;
       memw[$a000:(y-1) * 320 + x] := (c shl 8) + c;
       Inc(Y,2);
     until y >= 202;
     Dec(y,2);
     mem[$a000:y * 320 + x] := random(2) * 160;
     Inc(X,2);
```

```
   until x >= 320;
  until port[$60] < $80; {CLOSEGRAPH; }
  asm  mov ax, 3
       int 10h
  end;GOTOXY(30,12);TEXTCOLOR(WHITE);write('PAUL IOAN
MUNTEAN 2006');readln;
end;
```

Problema 2.Meniul cu altgoritmi speciali

Codul sursă:

Desenarea aleatoare pe ecran cu ajutorul tastaturi

```
procedure desenare;
 var X:Word:=40;
    Y:Word:=12;

 procedure SetCursor(Cursor: Word); assembler;
     asm
       mov     ax,$0100
       xor     bx,bx
       mov     cx,[Cursor]
       int     $10
 end;

 procedure HideCursor;
 begin
  SetCursor($2000);
 end;

 procedure ShowCursor;
 begin
  SetCursor($0607);
 end;

 begin
  CLRSCR;
  HideCursor;
  Writeln(' Tastele sageti: move & erase(deplasare)');
  Writeln(' Space         : drawing(desenare)');
  Writeln(' Ecs           : exit(Iesire');
```

```
 MultiKeysInit;
 repeat
  GotoXY(X,Y);
  if TestKey(Space_Scan) then Write('*') else Write(' ');
  if (TestKey(Right_Scan)) and (X<79) then inc(X);
  if (TestKey(Left_Scan)) and (X>1) then dec(X);
  if (TestKey(Down_Scan)) and (Y<24) then inc(Y);
  if (TestKey(Up_Scan)) and (Y>4) then dec(Y);
  GotoXY(X,Y);
  Write('Û');
  Delay(75);
 until TestKey(ESC_Scan);
 ShowCursor;
 MultiKeysDone;clrscr;
end;
```

Problema 3.Meniul cu altgoritmi speciali

Codul sursă:

```
Joc snake asemanator cu cele de pe nokia 3310
procedure snake;
 {$ifndef __CON__}
  This program must be compiled for MS-DOS, OS/2 or Win32 console mode
{$endif}

{$ifdef __WIN32__}
{$r snake.res}
{$endif}
{$ifdef __WIN32__}
{$r snake.res}
{$endif}

const
  MaxSections = 255;
  ScrDX       = 80;
  ScrDY       = 50;

var
  HeadX, HeadY: LongInt;
  SnakeX      : array [0..MaxSections] of Longint;
```

```
  SnakeY      : array [0..MaxSections] of Longint;
  Position    : LongInt;
  Sections    : LongInt;
  Direction   : LongInt;
  Screen      : array [1..ScrDX, 1..ScrDY] of Char;
  Score       : LongInt;

  isExit      : Boolean;
  Answer      : Char;
  DelayVal,i  : LongInt;
  SS, SE      : String := '';

  ScrHeight   : LongInt;
  ScrWidth    : LongInt;

  Key         : Char;
  OrigMode    : Word;

procedure PlayWave(SndNo, Flags: DWord);
begin
{$ifdef __WIN32__}
  MMSystem.PlaySound(MAKEINTRESOURCE(SndNo), HInstance, flags or
SND_RESOURCE);
{$endif}
end;

procedure DrawWindow(X, Y, DX, DY: LongInt);
var
  i: LongInt;
begin
  GotoXY(X, Y); Write('É', Copy(SE, 1, DX-1), '»');
  GotoXY(X, Y + DY); Write('È', Copy(SE, 1, DX-1), '¼');
  for i := Y + 1 to Y + DY-1 do
  begin
    GotoXY(X,i);
    Write('º');
    Write(Copy(SS, 1, DX - 1));
    Write('º');
  end;
```

```
end;

procedure PutSymbol(X, Y: Longint; Symbol: Char);
begin
  if (X in [1..ScrDX]) and (Y in [1..ScrDY]) then
  begin
    GotoXY(X, Y);
    Write(Symbol);
    Screen[X, Y] := Symbol;
  end;
end;

procedure DrawArea;
var
  i: Longint;
begin
  TextBackGround(Blue);
  ClrScr;
  FillChar(Screen, SizeOf(Screen), ' ');
  TextColor(White);
  for i := 2 to ScrWidth-1 do
  begin
    PutSymbol(i, 2, 'Í');
    PutSymbol(i, ScrHeight - 1, 'Í');
  end;
  for i := 3 to ScrHeight-1 do
  begin
    PutSymbol(1, i, 'º');
    PutSymbol(ScrWidth, i, 'º');
  end;
   PutSymbol(1, 2, 'É');
   PutSymbol(ScrWidth, 2, '»');
   PutSymbol(1, ScrHeight - 1, 'È');
   PutSymbol(ScrWidth, ScrHeight - 1, '¼');
   TextBackGround(White);
   TextColor(Black);
   GotoXY(1, 1); Write(Copy(SS, 1, ScrWidth));
  if ScrWidth > 40 then
  begin
```

```
    GotoXY(ScrWidth div 2 - 24,1);
    TextColor(Red);
    Write(' THE SNAKE GAME, ');
    TextColor(Black);
    Write('Copyright (c) 2006 by MASTERMIND&CO MUNTEAN.');
  end else
  begin
    GotoXY(ScrWidth div 2 - 7,1);
    TextColor(Red);
    Write('THE SNAKE GAME');
  end;
end;

procedure InitSnake(X, Y: LongInt);
var
  i: LongInt;
begin
  TextBackGround(Blue);
  TextColor(Yellow);
  for i := 0 to Sections - 1 do
  begin
    SnakeX[i] := i + X;
    SnakeY[i] := Y;
    PutSymbol(X + i, Y, 'P');
  end;
  HeadX := X;
  HeadY := Y;
end;

procedure InitBonus(num: LongInt);
var
  i, x, y: LongInt;
begin
  TextColor(White); TextBackGround(Blue);
  for i := 1 to num do
  begin
    repeat
      x := Random(ScrWidth - 3) + 2;
      y := Random(ScrHeight - 3) + 3;
```

```
    until Screen[x, y] = ' ';
    TextColor(Random(5) + 10);
    case Random (4) of
      1: PutSymbol(x, y, '□');
     2: PutSymbol(x, y, '');
      3: PutSymbol(x, y, '□');
    else
      PutSymbol(x, y, '□');
    end;
  end;
  TextColor(Yellow);
end;

procedure ViewScore;
begin
  GotoXY(ScrWidth div 2 - 14, ScrHeight);
  Write('[Score: ', Score:4, '   Sections: ', Sections:4, ']');
end;

procedure KillSnake;
const
  cr: array [0..5] of Char = ('X', 'x', '%', ':', '.' ,' ');
var
  i, j: LongInt;
begin
  case Random(3) of
    1:   PlayWave(107, 1);
    2:   PlayWave(104, 1);
    else PlayWave(103, 1);
  end;
  for j := 0 to 5 do
  begin
    for i := 0 to Sections - 1 do
    begin
      GotoXY (SnakeX[i], SnakeY[i]);
      if ((SnakeX[i] <> HeadX) or (SnakeY[i] <> HeadY)) and
         (SnakeX[i] <> 0) then Write(cr[j]);
    end;
    Delay(50);
```

```
  end;
  Delay(200);
end;

procedure AddSection;
begin
  if Sections < MaxSections then
  begin
    inc(Sections);
    SnakeX[Sections] := 0;
  end;
  ViewScore;
end;

procedure MoveSnake;
begin
  PutSymbol(SnakeX[Position], SnakeY[Position], ' ');
  case Direction of
    1:  Dec(HeadX);
    2:  Inc(HeadX);
    3:  Dec(HeadY);
    4:  Inc(HeadY);
  end;
  SnakeX[Position] := HeadX;
  SnakeY[Position] := HeadY;
  if Screen[HeadX, HeadY] in ['□', '', '□', '□'] then
  begin
    case Screen[HeadX, HeadY] of
      '□': PlayWave(105, 1);
      '': PlayWave(106, 1);
      '□': PlayWave(102, 1);
      '□': PlayWave(108, 1);
    end;
    inc(Score);
    AddSection;
    InitBonus(1);
  end else
  if Screen[HeadX, HeadY] <> ' ' then
  begin
```

```
    KillSnake;
    isExit := TRUE;
    exit;
  end;
  PutSymbol(SnakeX[Position], SnakeY[Position], '□');
  Dec(Position);
  if Position < 0 then Position := Sections - 1;
end;

begin
  FillChar(SnakeX, SizeOf(SnakeX), 0);
  FillChar(SnakeY, SizeOf(SnakeY), 0);
  DirectVideo := FALSE;
  CheckBreak := FALSE;

{$ifdef __WIN32__}
  SetConsoleTitle(' Snake Game - Win32 Console Application Demo');
{$endif}

  OrigMode := LastMode;
  TextMode(CO80 or Font8x8);

  HideCursor;

  ScrHeight := Hi(WindMax) + 1;
  ScrWidth  := Lo(WindMax) + 1;
  for i := 1 to 254 do
  begin
    SE := SE + 'Í';
    SS := SS + ' ';
  end;

  repeat
    DrawArea;

    TextColor(White);
    TextBackGround(Red);

    DrawWindow(ScrWidth div 2 - 16, ScrHeight div 2 - 2, 34, 8);
```

```
GotoXY(ScrWidth div 2 - 8, ScrHeight div 2 - 2);
Write(' Choose Game Level ');
GotoXY(ScrWidth div 2 - 14, ScrHeight div 2);
Write(' 1.Novice (Incepator) ');
GotoXY(ScrWidth div 2 - 14, ScrHeight div 2 + 1);
Write(' 2.Easy (Usor)? ');
GotoXY(ScrWidth div 2 - 14, ScrHeight div 2 + 2);
Write(' 3.Medium (Mediu) ');
GotoXY(ScrWidth div 2 - 14, ScrHeight div 2 + 3);
Write(' 4.Hard (Greu)!');
GotoXY(ScrWidth div 2 - 14, ScrHeight div 2 + 4);
Write(' 5.Expert (Foarte Greu)!');

PlayWave(101, 9);

repeat
  Answer := ReadKey;
  if Answer = #27 then
  begin
    ShowCursor;
    NormVideo;
    ClrScr;
    TextMode(OrigMode);
    Halt(0);
  end;
until Answer in ['1'..'5'];

case Answer of
  '1': DelayVal := 100;
  '2': DelayVal := 80;
  '3': DelayVal := 60;
  '4': DelayVal := 40;
  '5': DelayVal := 20;
end;

Position  := 3;
Sections  := 4;
Direction := 1;
isExit    := FALSE;
```

```
isExit    := FALSE;
Score     := 0;

DrawArea;
InitBonus(5);
InitSnake(ScrWidth div 2 - 2, ScrHeight div 2);
ViewScore;

PlayWave(106, 1);

repeat
 if KeyPressed then
 begin
   Key := ReadKey;
   if Key = #0 then
     case ReadKey of
      #72: if Direction <> 4 then Direction := 3;
      #80: if Direction <> 3 then Direction := 4;
      #75: if Direction <> 2 then Direction := 1;
      #77: if Direction <> 1 then Direction := 2;
     end;
   isExit := Key = #27;
 end;
 Delay(DelayVal);
 MoveSnake;

until isExit;

TextColor(White);
TextBackGround(Black);
DrawWindow(ScrWidth div 2 - 10, ScrHeight div 2 - 2, 20, 7);
TextColor(LightCyan);
GotoXY(ScrWidth div 2 - 4, ScrHeight div 2);
Write('Game Over');
TextColor(White);
GotoXY(ScrWidth div 2 - 8, ScrHeight div 2 + 2);
Write('Play again (Y/N)?');
gotoxy(ScrWidth div 2-8,scrheight div 2+3);
write('Joc Nou!');
```

```
    repeat
      Answer := UpCase(ReadKey);
    until (Answer = 'Y') or (Answer = 'N');

  until Answer = 'N';

  ShowCursor;

  NormVideo;
  ClrScr;
  TextMode(OrigMode);
end;
```

Problema 4.Meniul cu altgoritmi speciali

Codul sursă:

Program de transformare a oricarui text in cod morse

```
procedure morse;
   procedure trans_cod_morse1;
      begin
   for i:=1 to length(s4) do
       begin
   for j:=1 to 46 do
    if s4[i]=v1[j].carac  then
    write('   ',v1[j].cod);
       end;
   end;
      procedure trans_cod_morse2;
      begin
   for i:=1 to length(s4) do
       begin
   for j:=1 to 46 do
    if S4[i]=v1[j].carac  then
    write(f,'    ',v1[j].cod);
       end;  WRITELN(F,' ');
   end;
   procedure citire;
   begin
 v1[1].carac:='A'; v1[1].cod:='.-';
```

```
v1[2].carac:='B';v1[2].cod:='-...';
v1[3].carac:='C';v1[3].cod:='-.-.';
v1[4].carac:='D';v1[4].cod:='-..';
v1[5].carac:='E';v1[5].cod:='.';
v1[6].carac:='F';v1[6].cod:='..-.';
v1[7].carac:='G';v1[7].cod:='--.';
v1[8].carac:='H';v1[8].cod:='....';
v1[9].carac:='I';v1[9].cod:='..';
v1[10].carac:='J';v1[10].cod:='.---';
v1[11].carac:='K';v1[11].cod:='-.-';
v1[12].carac:='L';v1[12].cod:='.-..';
v1[13].carac:='M';v1[13].cod:='--';
v1[14].carac:='N';v1[14].cod:='-.';
v1[15].carac:='O';v1[15].cod:='---';
v1[16].carac:='P';v1[16].cod:='.--.';
v1[17].carac:='Q';v1[17].cod:='--.-';
v1[18].carac:='R';v1[18].cod:='.-.';
v1[19].carac:='S';v1[19].cod:='...';
v1[20].carac:='T';v1[20].cod:='-';
v1[21].carac:='U';v1[21].cod:='..-';
v1[22].carac:='V';v1[22].cod:='...-';
v1[23].carac:='W';v1[23].cod:='.--';
v1[24].carac:='X';v1[24].cod:='-..-';
v1[25].carac:='Y';v1[25].cod:='-.--';
v1[26].carac:='Z';v1[26].cod:='--..';
v1[27].carac:='0';v1[27].cod:='-----';
v1[28].carac:='1';v1[28].cod:='.----';
v1[29].carac:='2';v1[29].cod:='..---';
v1[30].carac:='3';v1[30].cod:='...--';
v1[31].carac:='4';v1[31].cod:='....-';
v1[32].carac:='5';v1[32].cod:='.....';
v1[33].carac:='6';v1[33].cod:='-....';
v1[35].carac:='7';v1[34].cod:='--...';
v1[34].carac:='8';v1[35].cod:='---..';
v1[36].carac:='9';v1[36].cod:='----.';
v1[37].carac:='.';v1[37].cod:='.-.-.-';
v1[38].carac:=',';v1[38].cod:='--..--';
v1[39].carac:=':';v1[39].cod:='---...';
v1[40].carac:='?';v1[40].cod:='..--..';
```

```
 v1[41].carac:=chr(41);v1[1].cod:='.----.';
 v1[42].carac:='-';v1[42].cod:='-....-';
 v1[43].carac:='/';v1[43].cod:='-..-.';
 v1[44].carac:=chr(40);v1[44].cod:='-.--.-';
 v1[45].carac:=chr(41);v1[45].cod:='-.--.-';
 v1[46].carac:=' ';v1[46].cod:='    ';
end;
  begin
    citire;    textbackground(BLUE);
    write('Codul Morse');
  for i:=1 to 46 do
     begin  textbackground(BLUE);
  writeln(v1[i].carac,'       ',v1[i].cod);
     end;READLN;clrscr;
  TEXTCOLOR(GREEN);WRITEln('Va rog itroduceti litere mari!!!');
  writeln('Dati caractele pe care  doriti sa le transformati in cod
morse');TEXTCOLOR(BROWN);
  WRITELN('REZULTATUL SE VA PUNE IN FISIERUL PAUL.DOC DIN
DIRECTORUL RADACINA');
  readln(s4);  trans_cod_morse1;
readln;
 ASSIGN(F,'paul.doc');rewrite(f);
 writeln(f,'Textul tastat:',s4);WRITEln(F,'Codul morse aferent textului:');
  trans_cod_morse2;
  writeln(f,'Codul morse');
    for i:=1 to 46 do
      begin
  writeln(f,v1[i].carac,'       ',v1[i].cod);
     end;READLN;close(f);
  end;
```

Problema 5.Meniul cu altgoritmi speciali

Codul sursă:

Afisarea unui hexagon 3D pe ecran

```
procedure hexagon3D;
```

```
{$r-,q-}
const
  pntsnr     = 12;
  facenr     = 20;
  polyvertnr = 3;        // nr of points in each polygon
  distance   = 4000;
  enddist    = 600;
  xcenter    = 0;
  ycenter    = 0;        // the pos of object on the screen
  zcenter    = 0;
  xrotadd    = 3;        // rotations about x,y,z axis
  yrotadd    = 4;
  zrotadd    = -2;
  ytopclip   = 0;
  ybotclip   = 200;
  norm       = 90;       // set this to -1000 to make the object transparent...

  // Conventional x,y,z format...
  Verts: array [1..(pntsnr*3)] of Longint=
    (200, 0, -120, 62, -189, -120, -160, -119, -120, -163, 115, -120,
     58, 191, -120, 0, 0, -243, -200, -3, 79, -62, -189, 80, -65, 188,
     80, 160, -115, 79, 160, 118, 78, -1, 0, 202);

  // Format: nr of faces; face1,face2,face3...; color;texmap info
  // This format is compatible with 3ded
  connect: array [1..facenr*6] of Longint=
    (3,  4, 3, 8,  6, 0, 3,  6, 7, 11,  5, 0, 3,  6, 2, 7,  7, 0 ,
     3,  2, 1, 7,  7, 0 ,3,  7, 1, 9,   5, 0, 3,  7, 9, 11, 6, 0,
     3,  11, 9, 10,  24, 0,  3,  9, 0,  10, 26 ,0,  3, 9, 1,  0, 7, 0,
     3,  5, 0, 1,  7, 0, 3,  5, 1, 2,  8, 0, 3,  6, 3, 2,  6, 0,
     3,  5, 2, 3,  6, 0, 3,  4, 0, 5,  7, 0, 3,  5, 3, 4,  6, 0,
     3,  10, 0, 4,  26, 0, 3,  8, 6, 11,  6, 0, 3,  10, 8, 11,  23, 0,
     3,  10, 4, 8,  25, 0, 3,  8, 3, 6,  6, 0);

type
  VecVerts = record
    x: Real;
    y: Real;
    z: Real;
```

```
end;

screen_coords = record
  x: DWORD;
  y: DWORD;
end;

var
  polyxyz: array[1..pntsnr] of vecverts;
  polyxyzb: array[1..pntsnr] of vecverts;
  normxyz: array[1..facenr] of vecverts;
  normxyzb: array[1..facenr] of vecverts;
  scrcoords: array[1..pntsnr] of screen_coords;
  rotxyz: array[1..pntsnr] of vecverts;
  rotnxyz: array[1..facenr] of vecverts;
  xr, yr, zr: Longint := 0;
  v, i, a, offs: Longint := 0;
  loop2: Longint := 1;
  Dist: Longint := 2000;
  VecCol: DWORD := 0;
  xadd, yadd, zadd: Real := 0;
  precsin: array[0..360] of Real;
  preccos: array[0..360] of Real;
  poly: array[0..199, 1..2] of Longint;
  Page: Boolean := FALSE;
  Mode3D: Boolean := TRUE;
  Ch: Char;

procedure FillMenu;
begin
  Rectangle(1, 1, 639, 479, 235);
  SetColor(120);
  OutTextXY(190, 400, 'Use "M" definirea view mode-lui');
  SetColor(215);
  OutTextXY(180, 420, 'Use "+" and "-" ajustarea distantei');
  SetColor(220);
  OutTextXY(260, 440, 'Use "ESC" EXIT ');
end;
```

```
procedure ChangePage;
begin
  if Page then
  begin
    SetActivePage(0);
    SetVisualPage(1, TRUE);
    Page := FALSE;
  end else
  begin
   SetActivePage(1);
   SetVisualPage(0, TRUE);
   Page := TRUE;
  end;
  SetFillColor(clBlack);
  Bar(160, 56, 480, 352);
end;

// This routine defines the vertices of the cube
// They must be in this order for calculating normals
procedure SetuPoints;
begin
  i := 1;
  v :=1 ;
  repeat
    polyxyzb[v].x := Verts[i];
    i := i + 1;
    polyxyzb[v].y := Verts[i];
    i := i + 1;
    polyxyzb[v].z := Verts[i];
    i := i + 1;
    v := v + 1;
  until v = pntsnr + 1;
  for i := 1 to facenr * 6 do
    connect[i] := connect[i] + 1
end;

// This creates the lookup table
procedure SetUpsc;
var
```

```
  loop1: Longint;
  ax, ay, az, bx, by, bz: Real;
begin
  offs := 0;
  for loop1 := 0 to 360 do
  begin
    precsin[loop1] := sin(loop1 * pi / 180);
    preccos[loop1] := cos(loop1 * pi / 180);
  end;
  offs := 0;
  for i := 1 to facenr do
  begin
    // Here I precalculate the normals and then all
    // I have to do is rotate them for each frame
    ax := (polyxyzb[connect[3 + offs]].x) - (polyxyzb[connect[2 + offs]].x);
    ay := (polyxyzb[connect[3 + offs]].y) - (polyxyzb[connect[2 + offs]].y);
    az := (polyxyzb[connect[3 + offs]].z) - (polyxyzb[connect[2 + offs]].z);
    bx := (polyxyzb[connect[4 + offs]].x) - (polyxyzb[connect[2 + offs]].x);
    by := (polyxyzb[connect[4 + offs]].y) - (polyxyzb[connect[2 + offs]].y);
    bz := (polyxyzb[connect[4 + offs]].z) - (polyxyzb[connect[2 + offs]].z);
    normxyz[i].x := 0;
    normxyz[i].y := 0;
    normxyz[i].z := 0;
    normxyzb[i].x := 0;
    normxyzb[i].y := 0;
    normxyzb[i].z := 0;
    normxyzb[i].x := (ay * bz) - (by * az);
    normxyzb[i].y := (az * bx) - (bz * ax);
    normxyzb[i].z := (ax * by) - (bx * ay);
    offs := offs + 6;
  end;
end;

procedure RotatePoints(xrot, yrot, zrot: Longint);
var
  sinxr, cosxr, sinyr, cosyr, sinzr, coszr: Real;
begin
  // here are all rotations values updated and precalculated
  xr := xr + xrot;
```

```
if xr > 360 then xr := 1;
if xr < 0 then xr := 360;
yr := yr + yrot;
if yr > 360 then yr := 1;
if yr < 0 then zr := 360;
zr := zr + zrot;
if zr > 360 then zr := 1;
if zr < 0 then zr := 360;
sinxr := precsin[xr];
cosxr := preccos[xr];
sinyr := precsin[yr];
cosyr := preccos[yr];
sinzr := precsin[zr];
coszr := preccos[zr];
for i := 1 to pntsnr do
begin
 // this is the formula for vector rotations around all axis in 3d space }
  rotxyz[i].x := 0;
  rotxyz[i].y := 0;
  rotxyz[i].z := 0;
  polyxyz[i].x := 0;
  polyxyz[i].y := 0;
  polyxyz[i].z := 0;
  polyxyz[i].x := polyxyzb[i].x;
  polyxyz[i].y := polyxyzb[i].y;
  polyxyz[i].z := polyxyzb[i].z;
  rotxyz[i].y := polyxyz[i].y * cosxr - polyxyz[i].z * sinxr;
  rotxyz[i].z := polyxyz[i].z * cosxr + polyxyz[i].y * sinxr;
  polyxyz[i].y := rotxyz[i].y;
  polyxyz[i].z := rotxyz[i].z;
  rotxyz[i].z := polyxyz[i].z * cosyr - polyxyz[i].x * sinyr;
  rotxyz[i].x := polyxyz[i].x * cosyr + polyxyz[i].z * sinyr;
  polyxyz[i].x := rotxyz[i].x;
  rotxyz[i].x := polyxyz[i].x * coszr - polyxyz[i].y * sinzr;
  rotxyz[i].y := polyxyz[i].y * coszr + polyxyz[i].x * sinzr;
end;
for i := 1 to facenr do
begin
  rotnxyz[i].x := 0;
```

```
   rotnxyz[i].y := 0;
   rotnxyz[i].z := 0;
   normxyz[i].x := normxyzb[i].x;
   normxyz[i].y := normxyzb[i].y;
   normxyz[i].z := normxyzb[i].z;
   rotnxyz[i].y := normxyz[i].y * cosxr - normxyz[i].z * sinxr;
   rotnxyz[i].z := normxyz[i].z * cosxr + normxyz[i].y * sinxr;
   normxyz[i].y := rotnxyz[i].y;
   normxyz[i].z := rotnxyz[i].z;
   rotnxyz[i].z := normxyz[i].z * cosyr - normxyz[i].x * sinyr;
   rotnxyz[i].x := normxyz[i].x * cosyr + normxyz[i].z * sinyr;
   normxyz[i].x := rotnxyz[i].x;
   rotnxyz[i].x := normxyz[i].x * coszr - normxyz[i].y * sinzr;
   rotnxyz[i].y := normxyz[i].y * coszr + normxyz[i].x * sinzr;
  end;
end;

procedure ScrProject(xc, yc, zc: Longint);
begin
  for i := 1 to pntsnr do
  begin
   scrcoords[i].x := trunc((320 * (rotxyz[i].x + xc) / (dist - (rotxyz[i].z + zc))) +
320);
   scrcoords[i].y := trunc(200 - (320 * (rotxyz[i].y + yc) / (dist - (rotxyz[i].z +
zc))));
  end;
end;

procedure Draw;
begin
  offs := -6;
  a := 1 ;
  for i := 1 to facenr do
  begin
   offs := offs + 6;
   VecCol := trunc(rotnxyz[i].z / 400);
   if rotnxyz[i].z > 10000 then
   begin
    SetColor(VecCol); SetFillColor(VecCol);
```

```
    if Mode3D then
      FillTriangle(scrcoords[connect[2 + offs]].x, scrcoords[connect[2 + offs]].y,
        scrcoords[connect[3 + offs]].x, scrcoords[connect[3 + offs]].y,
        scrcoords[connect[4 + offs]].x, scrcoords[connect[4 + offs]].y)
    else
      Triangle(scrcoords[connect[2 + offs]].x, scrcoords[connect[ 2+ offs]].y,
        scrcoords[connect[3 + offs]].x, scrcoords[connect[3 + offs]].y,
        scrcoords[connect[4 + offs]].x, scrcoords[connect[4 + offs]].y);
   end;
  end;
end;

// Here is the main program
begin
  // global variables initialized
  loop2 := 1;
  SetuPoints;                          // define vector vertices
  SetupSc;
  Dist := 2000;
  SetSVGAMode(640, 480, 8, LFBorBanked);
  if GraphResult <> 0 then
  begin
    ClrScr;
    Writeln(GraphErrorMsg(GraphResult));
    exit;
  end;
  DrawBorder := FALSE;
  ClearDevice;
  SetColor(125);
  OutTextXY(270, 230, 'Please wait...');
  SetActivePage(0);
  FillMenu;
  SetActivePage(1);
  FillMenu;
  for i := 1 to 126 do
    SetRGBPalette (i, i div 5, i div 3, (20 + i) div 3);
  repeat
    if dist < enddist then dist := enddist;
    rotatepoints(xrotadd, yrotadd, zrotadd);   // rotate around x,y,z axis
```

```
   scrproject(xcenter, ycenter, zcenter);
   Draw;
   ChangePage;
   if KeyPressed then
   begin
     ch := ReadKey;
     if UpCase(ch) = 'M' then Mode3D := not Mode3D;
     if (ch = '+') then Dist -:= 50;
     if (ch = '-') then Dist +:= 50;
      if Dist > Distance then Dist := Distance;
     if Dist < EndDist then Dist := EndDist;
   end;
  until ch = #27;
  CloseGraph;
end;
```

Problema 1.Meniul cu stringuri

Codul sursă:

{Pentru codificarea mesajelor transmite intre 2 terminale,se inlocuieste orice succesiune de mai multe caractere de acelasi fel cu un caracter de control # urmat de caracterul respectiv si nr lui de repetari succesive.Sa se scrie programul de codificare.

EX:abbbbxyyybbbzttt 0#b4x#y3#b3z#t3}

```
procedure eliminareLITrepetate;
begin
writeln('Pentru codificarea mesajelor transmite intre 2 terminale,se inlocuieste
orice');
writeln('succesiune de mai multe caractere de acelasi fel cu un caracter de
control #');
writeln('urmat de caracterul respectiv si nr lui de repetari succesive.Sa se
scrie');
writeln('programul de codificare.                    ' );
writeln('     EX:abbbbxyyybbbzttt    0#b4x#y3#b3z#t3');
write('Dati cuvantul:');readln(ar);
i:=1;bs:=' ';
repeat
  crs:=ar[1];
  nr:=0;
```

```
  while pos(crs,ar)<>0 do
       begin
       inc(nr);
       delete(ar,pos(crs,ar),1);
       end;
  str(nr,ss); {TRANSFORMA PE NR(UN NUMAR) INTR-UN STRING}
  bs:=bs+'#'+crs+ss;
until length(ar)=0;
writeln(bs);readln;
end;
```

Problema 2.Meniul cu stringuri

Codul sursă:

```
{Se da un cuvant format din maxim 255 de caractere. }
{Din acest cuvant se obtine parola sa prin scrierea de la dreapta }
{la stanga acaracterelor de pe pozitii impare.}
{Exemplu: pentru cuvantul algoritm parola este trga}
procedure parola;
var cuv:string;
function invers(cuv:string):string;
var    i,n:byte;
   l:string;
begin
l:='';
i:=1;
while i<=length(cuv) do
begin
l:=cuv[i]+l;
i:=i+2;
end;
invers:=l;
end;
begin
clrscr;
textbackground(12);
textcolor(11);
writeln('Se da un cuvant format din maxim 255 de caractere. ');
writeln('Din acest cuvant se obtine parola sa prin scrierea de la dreapta ');
writeln('la stanga acaracterelor de pe pozitii impare.');
```

```
writeln('Exemplu: pentru cuvantul algoritm parola este trga ' );
write('dati cuv ');read(cuv);
write('parola este ',invers(cuv));
readln;
readln;
end;
```

Problema 3.Meniul cu stringuri

Codul sursă:

```
{Sa se afiseze cuvintele continute de o fraza in ordine alfabetica.Fraza se
gaseste pe prima linie a fisierului fraza.in terminata cu '.',iar cuvintele
sunt separate prin 1 sau mai multe spatii}
procedure p7;
var f:text;
   a,aux:string;
    k,n,i,j,nr:byte;
begin
writeln('Sa se afiseze cuvintele in ordine alfabetica.Fraza se');
writeln('se va scrie de utilizator cuvant cu cuvant si se va termina cu  . iar
cuvintele');
writeln('sunt separate prin 1 sau mai multe spatii');
writeln('Dati nr de cuvinte din care este alcatuita fraza');readln(n);
writeln('Dati fraza!!!');
for i:=1 to n do
begin
writeln('Dati cuvantul  ',i,'-al frazei');
readln(cuv[i]);
end;
for i:=1 to n-1 do
  for j:=i+1 to n do
        if cuv[i]>cuv[j] then begin
                        aux:=cuv[i];
                        cuv[i]:=cuv[j];
                        cuv[j]:=aux;
                        end;
   writeln('Fraza sortata alfabetic este');
for i:=1 to n do write(cuv[i],' ');readln;
end;
```

Problema 4.Meniul cu stringuri

Codul sursă:

```
{se citeste caracter cu caracter un sir de caractere pana se introduce $ apoi se
afiseaza in ordine inversa cifrele din sir}
procedure cuvant;
var s:string;n:integer;
procedure cuv;
begin
for i:=n downto 1 do
if   ((s[i]='0')or(s[i]='1')or(s[i]='2')or(s[i]='3')or(s[i]='4')or(s[i]='5')
or(s[i]='6')or(s[i]='7')or(s[i]='8')or(s[i]='9')) then begin write(s[i]) end;
end;
begin
writeln('Se citeste caracter cu caracter un sir de caractere ');
writeln(' apoi se afiseaza in ordine inversa cifrele din sir');
write('Dati cuvantul: ');readln(s);n:=length(s);
cuv;readln;
end;
```

Problema 5.Meniul cu stringuri

Codul sursă:

Generarea tuturor prefixelor unui nume dat de la tastatura

```
procedure prefix;
var prefix,s:string;
l,i:integer;
begin
write(' Dati cuvantul ');readln(s);
l:=length(s);
writeln('Prefixurile cuv ',s,' sunt: ');
for i:=1 to l-1 do
begin
prefix:=copy(s,1,i);writeln(prefix);
end;readln;
end;
```

Programul poate fi îmbunătăţit prin adăugarea de noi probleme sub forma unor proceduri sau functii, pentru lărgirea softului şi pentru a putea răspunde unor cerinţe cât mai diverse.

Vă urăz succes în utilizarea acestui soft şi vă doresc să obţineţi performanţe cât mai mari in informatica.

Cap 1.Corpul principal al programului.

- **Programul principal de unde este condus tot programul**

Cap 2.Unitul ‘Ates ‘ cu sursele programelor.

- **Meniul grafic**
- **Meniul cu altgoritmi matematici**
- **Meniul cu fisiere**
- **Meniul cu grafuri**
- **Meniul BKTR**
- **Meniul Divide et Impera**

- **Meniul cu altgoritmi speciali**
- **Meniul cu stringuri**