TEHNICA GREEDY

Proiect creat de Postolache Alexandrina

ESENTA:

-se initializeaza multimea B la multimea vida;

-se alege un anumit element din multimea A;

-se verifica daca elementul ales poate fi adaugat la multimea B;

-procedeul continua pana ce au fost determinate toate elementele multimii B.

PROBLEMA RUCSACULUI

PROGRAM RUCSAC;

```
Var     g:array [1..10] of integer;
          i,n,Gm,R, aux : integer;
         ok:boolean;

begin
 writeln('nr obiecte'); readln(n);
writeln(‘capacitate  rucsac'); readln(R);
 writeln(' Obiectele de luat în rucsac:' );
 for  i:=1 to n do
        read (g[i]);
```

```
ok:=false;
while(ok=false) do
    begin
    ok:=true;
    for i:=1 to n-1 do
    if g[i]>g[i+1]   then
          begin
           aux:=g[i];
           g[i]:=g[i+1];
           g[i+1]:=aux;
           ok:=false;
          end;
  end;
```

{SORTAREA VECTORULUI}

```
writeln;
      for  i:=1 to n do
         write( g[i], '*');
Gm:=0 ;
  i:=1;
     while ( Gm +g[i]<=R ) do
     begin
      Gm:=Gm+g[i];
      i:=i+1;
     end;
  writeln('sunt‘ ,i-1,‘ obiecte cu greutate‘ , Gm,‘) ;
   writeln ( ‘  a ramas‘ , R-Gm,‘ loc liber‘ ) ;
end.
```

{ VERIFICĂ DACĂ FIECARE OBIECT ÎNCAPE ÎN RUCSAC }

PROBLEMA BANILOR

PROGRAM BANI;

```
type tablou=array[1..3,1..7] of integer;
var s,ss,i : integer; a:tablou; f:text;
{In primul rind al tabelului vom pastra nominalul bancnotelor}
{In al doilea rind - numarul bancnotelor citite din fisier}
{In al treilea rind - numarul bancnotelor obtinute la schimb}
Procedure Afisare(sa:integer);
    begin
    writeln('suma ',s);
    if sa<>0 then
        writeln('nu poate fi transformata cu bancnotele date ‘)
    else begin
        writeln('se plateste cu urmatoarele bancnote’);
    for i:=1 to 7 do
        if a[3,i]<>0 then
            writeln('bancnote de ',a[1,i]:6,' sau folosit ',a[3,i]);
        end;
    end; { Afisare }
```

```
Procedure calcul(var sa:integer);
var nb:integer;
begin
i:=7;
while (i>=1) and (sa>0) do
    begin
    nb:=sa div a[1,i];
    if nb<>0 then if nb>= a[2,i]
        then a[3,i]:=a[2,i]
    else a[3,i]:=nb;
    sa:=sa-a[3,i]*a[1,i];
    i:=i-1;
    end;
end; { calcul }
```

```
begin
a[1,1]:=1;
a[1,2]:=5;
a[1,3]:=10;
a[1,4]:=50;
a[1,5]:=100;
a[1,6]:=200;
a[1,7]:=500;
assign (f,'bani.in');
reset(f);
for i:=1 to 7 do readln(f,a[2,i]);
write ('introduceti suma de lei S ');readln(s);
ss:=s; calcul(ss); Afisare(ss);
end.
```

PROBLEMA SPECTACOLELOR

PROGRAM SPECTACOL;

```
type teatru=record
ins, sfs:integer; (ora de inceput si de sfarsit a unui spectacol
calculata in minute scurse fata de miezul noptii)
ord:integer; (numarul de ordin al spectacolului)
end;
Var v:array [1..30] of teatru;
n, ultim, nr:integer; (n=numarul de spectacole, in variabila ultim
avem in permanenta ultimul spectacol selectat, nr=numarul maxim
de spectacole)

Procedure sortare_piese;
Var i,j:integer;
    temp:teatru;
```

```
Begin
For i:=1 to n-1 do
for j:=i+1 to n do
if v[j].sfs<v[i].sfs then
begin
temp:=v[i];
v[i]:=v[j];
v[j]:=temp;
end;
Procedure citire_piese;
Var hh,mm,i:integer;
begin
Write (‘Numarul de piese de teatru n= ‘);
Readln (n);
for i:=1 to n do begin
Write (‘Piesa cu nr ‘,i, ‘cand incepe?
(ora si minutul)’);
Readln (hh,mm);
v[i].ins:=hh*60+mm;
Write (‘Piesa cu nr ‘,i, ‘cand se termina?
(ora si minutul)’);
Readln (hh,mm);
v[i].ins:=hh*60+mm;
v[i].ord:=i;
end; end;
```

```
Procedure afis_piese;
Var i:integer;
Begin
Write (‘Inceputurile si sfarsiturile pieselor in minute scurse de la miezul noptii: ‘);
for i:=1 to n do
write (‘(‘,v[i].ins,’,’,v[i].sfs,’,’,v[i].ord,’)’);
writeln;
end;
```

```
Procedure algo_greedy;
Var i:integer;
Begin
Write (‘Piesele posibile, in ordine: ‘);
ultim:=1; nr:=1;
write (v[i], ‘ ‘);
for i:=2 to n do
If (v[i].ins>v[ultim].sfs) then
Begin
Write (v[i].ord, ‘ ‘);
ultim:=i;
nr:=nr+1; end;
```

```
Writeln (‘In total se pot alege maxim’,nr,’ piese’);
end;
Begin
citire_piese;
afis_piese;
sortare_piese;
afis_piese;
algo_greedy;
end.
```

PROBLEMA BENZINARIEI

PROGRAM BENZINARIE;

```
Type benz=record
ins, sfs:integer;
ord:integer; end;
Var v:array [1..100] of benz;
n, ultim, nr:integer;
Procedure citire_clienti;
Var hh, mm, i:integer;
begin
Write (‘n= ‘); Readln (n);
for i:=1 to n do begin
Write (‘Clientul cu nr. ‘,i,’cand este servit? (ora si minutul)’);
Readln (hh, mm);
v[i].ins:=hh*60+mm;
Write (‘clientul cu nr ‘, i, ‘ cand a terminat alimentarea ? ‘);
Readln (hh, mm);
v[i].sfs:=hh*60+mm;
v[i].ord:=i; end; end;
Procedure afisare_clienti;
Var i:integer;
Begin
Write (‘ cand incepe sa fie servit si cand a terminat alimentarea: ‘);
for i:=1 to n Do
Write (‘(‘v[i].ins,’,’,v[i].sfs, ‘,’,v[i].ord’)’);
Writeln;  end;
```

```
Procedure sortare_clienti;
Var i,j:integer;
t:benz;
Begin
for i:=1 to n-1 Do
for j:=i+1 to n Do
if (v[j].sfs<v[i].sfs) then
Begin
t:=v[i]; v[i]:=v[j];
v[j]:=t; end; end;
```

```
Procedure alg_greedy;
var i:integer;
Begin
Write ('posibilii clienti, in ordine: ');
ultim:=1;
nr:=1;
Write (v[i].ord, ' ');
for i:=2 to n do
if (v[i].ins>v[ultim].sfs) then
begin
Write (v[i].ord, ' ');
ultim:=i;
nr:=nr+1;
end;
```

```
Writeln ('in total se pot alege maxim',nr, 'clienti');
begin
citire_clienti;
afisare_clienti;
sortare_clienti;
afisare_clienti;
alg_greedy;
end.
```

PROBLEMA MAXIMULUI

PROGRAM MAXIM;

```
Var n, a1, a2, c:Integer;
Begin
a1:=-MAXINT;  (initializam primele 2 numere si n cu o constanta predefinita)
a2:=-MAXINT;
n:=-MAXINT;
While n<>0 Do Begin
If (n>a1) Then a1:=n;  (daca numarul n este mai mare decat primul cel mai mare numar atunci maximul este n)
If (a2<a1) Then Begin
c:=a1;
a1:=a2;
a2:=c;  end; (interschimbare)
Readln (n); end;
Writeln ('a1, ' ',a2');
end.
```

BIBLIOGRAFIE

- http://dasinika.blogspot.com/2009/04/tehnica-greedy-pentru-problemele-pentru.html
- http://paulborza.blogspot.com/2010/08/metoda-greedy-in-limbajul-de-programare.html
- https://tpascalblog.wordpress.com/
- https://www.slideshare.net/yoanna_ioana/problema-rucsacului-presentation-948687

MULTUMESC PENTRU ATENTIE!