Lucrare de control -informatică –nr1

I. Răspundeți următoarelor cerințe:

1)Ce va afișa următorul program:

```
var a:array[1..3,1..3] of integer;
    i,j:integer;
begin
for i:=1 to 3 do
  for j:=1 to 3 do
   a[4 -i,j]:=i+j;
for i:=1 to 3 do
  for j:=3 downto 1 do
    write(a[i,j])
end.
```

a) 456345234
b) 234345456
c) 432543654
d) 654543432

.....

Se consideră un graf neorientat ,cu n noduri ,reprezentat prin matricea de adiacență a.

II.Modificați următoarea secvență reprezentată prin pseudocod astfel încât să determine gradul maxim al vârfurilor din graful dat .Matricea de adiacență se consideră completată.Explicați semnificația vectorului d și a variabilei m.

{1} pentru i=1 la n execută
{2} pentru j=i la n execută
{3}dac[a[I,j]=1 atunci
{4}d[i]<-d[i]+1
{5}d[j]<-d[i]
Sf dacă
{6}m<-0
{7}pentru k=2 la n execută
{8}dacă d[k]<m atunci
{9} m<-d[k]

III.Parcurgeți în lățime graful din figură .

IV. Se consideră graful cu n=4 şi u={(1,2),(1,3),(2,3),(3,4),(1,4)}.Câte grafuri parțiale conexe are graful ?

V.Să se scrie procedura backtracking pentru aflarea ciclurilor de lungime k și procedura de afișare a soluțiilor într-un fișier text ,fiecare soluție pe un rând.Să se explice semnificația variabilelor folosite.

VI.Să se scrie programul pentru verificarea conexității unui graf neorientat în care se cunosc muchiile . În cazul în care graful nu este conex să se determine numărul componentelor conexeRezultatul să fie afișat într-un fișier text .

Ex1-0,25 Ex2-1,25 Ex3-0,5 ex4-0,5 ex5-2,5 ex6-3p

......

Lucrare de control -informatică –nr2

I. Răspundeți următoarelor cerințe:

1. Ce va afișa următorul program:

```
var s:string[4]; i,j:byte; c:char;
begin
s:='anRB';
for i:=1 to 3 do
   for j:=1 to 4-i do
     if s[j]>s[j+1]
       then begin
            c:=s[j]; s[j]:=s[j+1];
s[j+1]:=c;
            write(s);
            end
end.
```

a) aRnBanBRaBnR
b) aRnBaRaRRaRBBR
c) aRnBRnaBBnaRBanRBRnaBRan
d) aRnBaRBnRaBnRBanBRan

……

Se consideră un graf neorientat ,cu n noduri ,reprezentat prin matricea de adiacență a.

II.Modificați următoarea secvență reprezentată prin pseudocod astfel încât să determine condițiile interne necesare validării unui element pentru un ciclu elementar ,al grafului dat .Matricea de adiacență se consideră completată.Explicați semnificația vectorului s și a variabilei k.

{1}v<-adevărat
{2} dacă a[s[k],s[k-1]]=1
{3}atunci v<-fals sf _dacă
{4} dacă v atunci
{5}pentru i=1 la k-1 do execută
{6}bloc dacă s[i]<>s[i+1]
{7}atunci v<-fals sf_dacă sf-bloc

III.Parcurgeți în lățime graful din figură .

IV.Se consideră un graf cu n=6 şi u={(1,2),(1,3),(6,5),(3,4),(4,5),(4,6)}Care este numărul maxim de muchii care pot fi eliminate pentru a se obține un garf parțial al grafului.

V. Să se scrie procedura bactracking pentru aflarea tuturor ciclurilor elementare care conțin un vârf dat v ,unde v e diferit de primul nod și este citit dint-un fișier text , și funcția valid .Să se explice semnificația variabilelor folosite.

VI.Să se scrie programul pentru descompunerea unui graf în componente conexe ,afișându-se componentele conexe într-un fișier text, și cunoscând matricea de adiacență de la tastatură.

Ex1-0,25 Ex2-1,25 Ex3-0,5 ex4-0,5 ex5-2,5 ex6-3p

Parcurgerea grafurilor neorientate –recursiv

Parcurgere BF-recursiv (în pp citesc start ,inițializez coada şi vectorul viz cu 0,pi=1, pun vf start în coadă şi îl vizitez ,p-poziția curentă la care s-a ajuns în coadă,în pp se începe de la poziția 1 din coadă deci BF(1)).

```
Procedure Bf (p:integer);
Begin
For  k:=1 to n do
If a[c[p],k]=1 and(viz[k]=0)  then begin
  Pi:=pi+1;
C[pi]:=k;
Viz[k]:=1;
End;
If p<=pi  then Bf(p+1);
End;
```

Parcurgerea DF- are ca parametru nodul curent asupra căruia se aplică parcurgerea.
Se afişează nodul asupra căruia se aplică şi-l marchează cu 1 ca fiind vizitat,apoi pentru fiecare vecin nevizitat de-al nodului curent se face autoapel asupra nodului curent.

```
Procedure Df (p:integer);
Begin
Write(start:4);viz[start]:=1;
For  k:=1 to n do
If a[c[start],k]=1 and(viz[k]=0)  then
  Df(k);
End;
```

Parcurgere df
Citesc nodul de plecare ,îll pun în coadă şi îl vizitez,apoi while pi:'1 do df; sfârşit .

```
Begin
Z:=c[pi];
K:=urm[z]+1;
While (k<=n )and((a[z,k]=0)or(a[z,k]=1))and(viz[k]=1) do k:=k+1;
Urm[z]:=k;
If k=n+1 then pi:=pi-1
Else begin
        Viz[k]:=1;pi:=pi+1;c[pi]:=k;
         Write(k:3);
End;
End;
```