## **VARIANTA 6 - Rezolvare**

## Subjectul 1 - rezolvare

1.	În cadrul expresiei se întâlnesc operatori relaționali și operatori aritmetici. Se ține cont de prioritatea operatorilor în evaluarea expresiei (operatorii aritmetici au proritate mai mare decât operatorii relaționali).  Pentru precizarea răspunsului a) se acordă 4p.	
2.	Fie <b>n</b> numărul de noduri. În acest caz numărul de muchii este <b>n-1</b> . Cum <b>n=1+1200</b> x $3 + 300$ x $2 + 1 = 4202$ rezultă că numărul de muchii este $4201$ . Pentru răspunsul <b>b</b> ) se acordă $(4p.)$	
3.	Un graf neorientat este complet dacă oricare două noduri sunt adiacente. Prin urmare un graf este complet dacă numărul total de muchii este <b>n(n-1)/2</b> . Pentru alegerea variantei de răspuns <b>d</b> se acordă <b>(6p.)</b>	
4.	Se generează în maniera "înaintare cu revenire", șirurile:	
	1010,1012,1014,1030,1032,1034, etc. Al patrulea număr generat este 1030. Pentru răspunsul b) se acordă (4p.)	
5.	La apelul rec(17,2) deoarece 2x2<17 se va afișa valoarea 2, variabila d va primi valoarea 3 și va avea loc apelul rec(17,4). Deoarece 4x4<17 se va afișa 4, variabila d va primi valoarea 5 și va avea loc apelul rec(17,6). Cum 6x6>17 nu va mai fi un nou apel și se vor afișa valorile din memorie în ordinea inversă memorării adică 5 și 3. Pentru răspunsul 2 4 5 3 varianta b se acordă (4p.)	

## **Subjectul 2 - rezolvare**

1.	a)	Se observă că șirul de valori citite este format din mai multe secvențe care se termină cu 0. Parcurgând pas cu pas algoritmul se observă că pentru fiecare secvență se determină elementul cu valoarea maximă. Pentru exemplu dat, prima secvență are elementul cu valoarea maximă 34, a doua are elementul cu valoarea maximă 100, iar ultima are elementul cu valoarea maximă 50. Deci se vor afișa valorile: 34,100,50. (6p).
	b)	Pentru afișarea aceleiasi valori de trei ori trebuie căutat un set de date de intrare care conține trei secvențe finalizate cu 0 care au același element cu valoare maximă. Un exemplu ar putea fi 33, 4, 5, 0, 7, 11, 33, 0, 23, 33, 0, 0. Pentru precizarea corectă a unui set de date conform cerinței se vor acorda 6p.
	c)	Structura cât timp execută poate fi înlocuită cu structura repetitivă repetă până când.Trebuie să avem în vedere că aceasta se execută cel puțin odată, deci verificăm dacă a>0 sau b>0. Prezentăm mai jos o posibilă soluție:

```
citeste a,b (numere naturale nenule)

m a

rdacă a>0 sau b>0 atunci

repeta

| a b

| citeste b

| rdacă a>m atunci

| m a

| m a

| m a

| m a

| pacă b=0 și a>0 atunci

| m a

| pacă b=0 și a>0 atunci

| pacă b=0 și a>0 atunci
```

Pentru integrarea corectă a unei structuri repetitive se acordă **3p**., pentru rescrierea integrală a algoritmului se acordă **3p**. - în total **6p**.

d) Pentru instrucțiunile corecte de declarare a variabilelor, de citire a datelor, de afișare a rezultatului și de decizie se acordă câte 3p., pentru scrierea corectă a instrucțiunii repetitive se acordă 3p., iar pentru atribuiri se acordă 3p. Pentru structura corectă a programului se mai acordă 1p., în total 10p.

O posibilă implementare în limbajul de programare studiat este:

Limbajul Pascal	Limbajul C/C++
<pre>var a,b,m:integer; begin     read(a,b);     m:=a;     while (a&gt;0) or (b&gt;0) do     begin     a:=b;     read(b);     if(a&gt;m) then         m:=a;     if (a&gt;0) and (b=0) then         begin         write(m,' ');         m:=a;     end; end; end.</pre>	<pre>using namespace std; int main() {   int m,a,b;   cin&gt;&gt;a&gt;&gt;b;    scanf("%d%d",&amp;a,&amp;b);   m=a;   while(a&gt;0  b&gt;0) {      a=b;      cin&gt;&gt;b;          scanf("%d",&amp;b);         if (a&gt;m) m=a;         if ((b==0)&amp;&amp;(a&gt;0))         {cout&lt;<m<<" ";<="" td=""></m<<"></pre>

Pentru scrierea corectă a unui set de instrucțiuni și pentru citirea variabilei **m** se acordă (**3p.**). Pentru scrierea corectă a instrucțiunii de decizie și afișarea conținutului variabilei **m**, conform cerinței, se acordă încă (**3p.**). Total (**6p.**)

Un exemplu de implementare ar putea fi:

Limbajul Pascal	Limbajul C/C++
<pre>readln(m.nume, m.datan.zi, m.datan.luna, m.datan.an); if(m.datan.an &gt; 2000) then</pre>	<pre>cin&gt;&gt;m.nume&gt;&gt;m.datan.zi &gt;&gt;m.datan.luna&gt;&gt;m.datan.an;  scanf("%s%d%d",&amp;m.nume, &amp;m.datan.zi,&amp;m.datan.luna, &amp;m.datan.an); if(m.datan.an &gt; 2000)cout&lt;<m.nume; "<<="" <<"="" cout<<m.datan.zi="" datan.="" else="" luna<="" m.="" td=""  printf("%s",m.nume);=""></m.nume;></pre>

Secvența de instrucțiuni menționată realizează următoarele modificări asupra șirului dat: dacă pe locul i se află un caracter literă mare acesta devine literă mică altfel caracterul de pe locul i se șterge. Prin ștergere, caracterul de pe locul i+1 ajunge pe locul i explicând astfel de ce nu se șterg toate caracterele litere mici și de ce rămân caractere litere mari neconvertite în litere mici. Prin urmare șirul BacaLauReat va deveni bcLura. Pentru rezolvarea corectă a acestei cerinte se acordă (6p.)

## Subjectul 3 - rezolvare

- 1. In cadrul subprogramului se parcurge tabloul unidimensional primit ca parametru, și se determină numărul elementelor cu valori pare din cadrul unei secvențe de **k** elemente consecutive. Se returnează poziția de început a secvenței cu număr maxim de valori pare sau **0** în cazul în care tabloul nu are elemente cu valori pare.
  - a) Pentru respectarea structurii antetului **function/int** se acordă **1p**., pentru declararea corectă a parametrilor de intrare **1p**, pentru respectarea structurii subprogramului și a sintaxei limbajului se acordă **1p**. Pentru determinarea secvenței cu cele mai multe numere pare se acordă **2p**. Pentru transmiterea corectă a rezultatului se acordă **1p**. In total (**6p**.)

Un exemplu de implementare poate fi:

Limbajul Pascal	Limbajul C/C++
<pre>function(n:integer;x:vector;k: integer):integer; var i,j,poz,max,s:integer; begin</pre>	<pre>int p(int n, int x[], int k) {    int max1=0,poz=-1,s=0,ok=0;    for (int i=0;i<n-k;i++) (int="" for="" j="i;j&lt;=i+k;j++)&lt;/pre" {=""></n-k;i++)></pre>
<pre>for i:=1 to n-k+1 do begin s:=0; for j:=i to k+i-1 do     if(x[j] mod 2=0)     then inc(s);</pre>	<pre>if (x[j]%2==0)     {s++; ok=1;}  if (s&gt;=max1)     {max1=s;     poz=i;} </pre>

```
if(s>=max) then
begin
    max:=s;
    poz:=i;
end;
end;
if(max>0) then
    p:=poz
else
    p:=0;
end;
if(ok)
return poz;
else return -1;
}
```

- a) Pentru declararea corectă a variabilelor globale se acordă **1p**, pentru citirea valorilor șirului de numere se acordă **1p**. pentru apelarea corectă a subprogramului se acordă **1p**. Pentru corectitudine globală se acordă **1p**. Total **(4p.)**
- Pentru corectă a citirea declararea variabilelor se acordă 1p., pentru variabilelor **n** și **m** se acordă 1p., pentru citirea tabloului bidimensional acordă 1p., pentru determinarea pozițiilor celor două linii care se vor interschimba, 2p, pentru algoritm corect de interschimbare linii se acordă 2p., iar pentru afișarea tabloului bidimensional se acordă 1p. În plus, pentru declararea variabilelor simple, structura și corectitudinea sintactică a programului se mai acordă 2p. În total 10p.

Un exemplu de implementare fi:

```
i := x1:
                                             cout<<v[i][j]<<" ";
                end;
                                             |printf("%d",v[i,j]);
                                        cout << endl;
        end;
        for j:=1 to m do
        begin
                                    return 0;
        aux := a[x1][j];
        a[x1][j]:=a[x2][j];
        a[x2][j] := aux;
        end;
        for i:=1 to n do
        begin
        for j:=1 to n do
       write( a[i][j], ' ');
        writeln;
        end;
end.
```

3. a) Se acordă **2p.** pentru operația de citire din fișier (declarare, nume corect și deschidere pentru citire), **2p.** pentru citirea tuturor numerelor din fișier, **2p.** pentru un algoritm corect de afișare a valorilor cerute, **2p.** pentru corectitudinea globală (declararea variabilelor, structura programului, sintaxa instrucțiunilor, etc.). Total **8** puncte.

Un exemplu de implementare poate fi:

Limbajul Pascal	Limbajul C/C++
<pre>var     f:text;     n,i,m,j,x:integer; begin     assign(f,'date.in');     reset(f);     readln(f,n,m);     for i:=1 to n do         for j:=1 to m do         begin</pre>	<pre>#include <iostream> #include <fstream> using namespace std; ifstream f("bac.txt");  FILE *f=fopen("bac.txt","r"); int main() {   int n,i,j,m,x;   f&gt;&gt;n&gt;&gt;m;    fscanf(f,"%d%d",&amp;n, &amp;m);    for(i=1;i&lt;=n;i++)       for(j=1;j&lt;=m;j++)       {       f&gt;&gt;x;        fscanf(f,"%d",&amp;v[i]);       if(x==1)       cout&lt;<i<<" "<<j<";";<="" pre=""></i<<"></fstream></iostream></pre>
end.	<pre>f.close(); return 0; }</pre>

Algoritmul este eficient deoarece pentru implementare nu se utilizează tablouri unidimensionale sau bidimensionale iar parcurgerea elementelor din fișier se face o singură dată. Se parcurge fișierul date. in și se afișează coordonatele fiecărui element nenul (secvența:i; locul în cadrul secvenței:j,unde i parcurge toate secvențele i=1, n iar j parcurge toate elementele unei secvențe j=1, m). Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. - în total 2p.