SUBIECTUL I **(20 de puncte)**

Pentru fiecare dintre itemii de la 1 la 5, scrieţi pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Indicați numerele pe care le pot memora variabilele întregi **x** şi **y**, astfel încât valoarea expresiei C/C++ alăturate să fie **1**.

```
x/2+x%y-x/y==0
```

a. **x=4** şi **y=2** **b.** **x=6** şi **y=3**

c. **x=8** şi **y=4** **d.** **x=10** şi **y=0**

2. Subprogramul **f** este definit alăturat. Valoarea lui **f(6)** este:

```
int f(int n)
{ if (n<=2) return n;
  if (n%2==1) return f(n-2)-f(n-1);
  return f(n-1)-f(n-2);
}
```

a. **6** **b.** **5** **c.** **2** **d.** **1**

3. Variabila **x** este de tip **char** şi memorează o literă mică a alfabetului englez. Indicați expresia C/C++ care are valoare nenulă dacă şi numai dacă litera memorată în variabila **x** este o vocală. Se consideră vocale literele **a**, **e**, **i**, **o**, **u**.

a. `strcmp(x,"aeiou")==0`

b. `strchr("aeiou",x)`

c. `'a'<=x && x<='u'`

d. `x==a || x==e || x==i || x==o || x==u`

4. Utilizând metoda backtracking, se generează, în ordine strict descrescătoare, toate numerele naturale de câte patru cifre distincte din mulţimea {**0**, **1**, **2**, **3**, **4**, **5**}. Primele şase numere generate sunt, în această ordine: **5432**, **5431**, **5430**, **5423**, **5421**, **5420**. Al şaptelea număr generat este:

a. **5415** **b.** **5413** **c.** **5342** **d.** **5340**

5. Un graf neorientat are **20** de noduri şi **10** muchii. Numărul maxim de componente conexe pe care le poate avea acest graf este:

a. **5** **b.** **10** **c.** **16** **d.** **20**

SUBIECTUL al II-lea **(40 de puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod.
S-a notat cu **a%b** restul împărţirii numărului natural **a** la numărul natural nenul **b**.

```
citeşte m,n,p,q
   (numere naturale nenule, p≤q)
 s1←0; s2←0
┌cât timp p≤q execută
│┌dacă p%m=0 sau p%n=0 atunci
│| s1←s1+1
│└■
│┌dacă p%m=0 şi p%n=0 atunci
│| s2←s2+1
│└■
│ p←p+1
└■
 s←s1-2*s2
 scrie s
```

a) Scrieţi valoarea afişată dacă se citesc, în această ordine, numerele **4**, **3**, **11** şi **25**. 3 **(6p.)**

b) Dacă pentru **m**, **n** şi **p** se citesc numerele **3**, **5**, respectiv **1**, scrieţi două numere care pot fi citite pentru **q** astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afişată să fie **10**. **(6p.)**

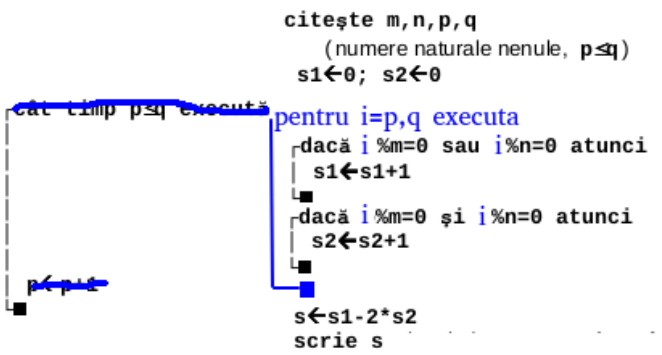
oricare doua din numerele 25,26,30,31,32

c) Scrieţi programul C/C++ corespunzător algoritmului dat. **(10p.)**

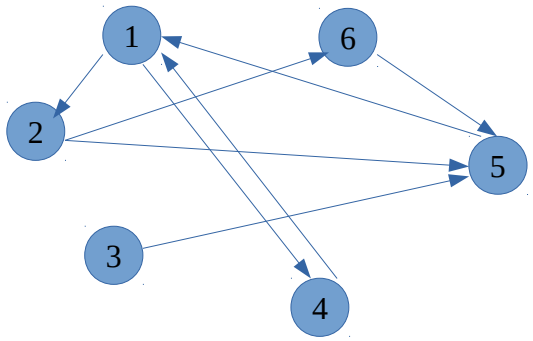
d) Scrieţi în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **cât timp...execută** cu o structură repetitivă de tip **pentru...execută**. **(6p.)**

c) si d)

```
#include<iostream>
using namespace std;
int main()
{
unsigned long m,n,p,q,s1,s2,s;
cin>>m>>n>>p>>q;
s1=0;s2=0;
while(p<=q){
        if(p%m==0 || p%n==0) s1=s1+1;
   if(p%m==0 && p%n==0) s2=s2+1;
   p=p+1;
}
s=s1-2*s2;
cout<<s;
return 0;}
```



2. Un graf orientat cu **6** vârfuri, numerotate de la **1** la **6**, are arcele **(1,2)**, **(1,4)**, **(2,5)**, **(2,6)**, **(3,5)**, **(4,1)**, **(5,1)**, **(6,5)**. Scrieţi un drum elementar de lungime maximă din graful dat. **(6p.)**



3, 5, 1, 2, 6

3. Variabila **fig**, declarată alăturat, memorează lungimea razei unui cerc și coordonatele centrului acestuia, în sistemul de coordonate **xOy**.
Scrieți o secvență de instrucțiuni prin care se inițializează variabila **fig**, astfel încât cercul corespunzător acesteia să aibă raza **1** și centrul în originea sistemului de coordonate. **(6p.)**

```
struct punct
{  float x,y;
};
struct cerc
{  struct punct centru;
   float raza;
}fig;
```

```
fig.raza=1;
fig.centru.x=0;
fig.centru.y=0;
```

SUBIECTUL al III-lea **(30 de puncte)**

1. Subprogramul **Impare** are un singur parametru, **n**, prin care primește un număr natural (**n∈[1,10⁹]**), cu cel puțin o cifră impară. Subprogramul înlocuiește fiecare cifră impară a lui **n** cu cea mai mare cifră pară strict mai mică decât ea (astfel cifra **1** se înlocuiește cu cifra **0**, cifra **3** cu cifra **2** etc.) și furnizează numărul obținut tot prin parametrul **n**. Scrieți definiția completă a subprogramului.
Exemplu: dacă **n=235690**, atunci, după apel, **n=224680**, iar dacă **n=15690**, atunci, după apel, **n=4680**. **(10p.)**

```
void Impare(unsigned long &n)
{
unsigned long m=0,p=1; int r;
while(n!=0)
{
        r=n%10;
        n=n/10;
        if(r%2!=0) r=r-1;
        m=m+p*r;
        p=p*10;
}
n=m;
}
```

2. Un tablou bidimensional cu număr impar de coloane este numit **simetric faţă de coloana din mijloc** dacă, pe fiecare linie a tabloului, elementele dispuse simetric faţă de elementul din mijloc al liniei respective au valori egale.
Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale din intervalul **[3,21]**, **m** şi **n** (**n** impar), şi elementele unui tablou bidimensional cu **m** linii şi **n** coloane, numere naturale din intervalul **[0,10⁴]**. Programul afişează pe ecran mesajul **DA**, dacă tabloul este simetric faţă de coloana din mijloc, sau mesajul **NU** în caz contrar.
Exemplu: pentru **m=4**, **n=5** și tabloul alăturat, se afişează pe ecran **DA** **(10p.)**

1	2	4	2	1
3	5	5	5	3
2	4	1	4	2
1	1	1	1	1

```
#include <iostream>
using namespace std;
int a[22][22],n,m;
void verifica()
{
	int i,j,da=1;
	for(i=0;i<m;i++)
		for(j=0;j<(n-1)/2;j++)
		  if(a[i][j]!=a[i][n-j-1]) da=0;
	if(da==0) cout<<"NU";
		else cout<<"DA";
}
void citeste()
{
	cin>>m>>n;
	for(int i=0;i<m;i++)
		for(int j=0;j<n;j++)
			cin>>a[i][j];
}
int main()
{
citeste();
verifica();
return 0; }
```

3. Un termen al unui șir de numere se numește **vârf local** al acestuia dacă nu există niciun alt termen mai mare sau egal cu el care să îl preceadă în șir sau dacă este egal cu termenul vecin anterior, iar acesta este vârf local.
Fișierul **bac.txt** conține un șir format din cel puțin două și cel mult 10^6 numere naturale din intervalul $[0,10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran, separate prin câte un spațiu, toate vârfurile locale ale șirului aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
Exemplu: dacă fișierul conține numerele **7 4 9 10 10 10 8 10 10 8 30**
se afișează pe ecran **7 9 10 10 10 30**
a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
b) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("bac.txt");
int main()
{
	int a,b,max,pozmax,n;
	f>>a; cout<<a<<' ';
	max=a;pozmax=0;n=1;
	while(f>>b)
	{
		if(b>max)
			{max=b;cout<<b<<' ';pozmax=n;}
		  else
			if(a==b && a==max && n==pozmax+1)
				{cout<<b<<' ';pozmax=n;}
		a=b;
		n=n+1;
	}
f.close();
}
```

Rolul variabilelor din program:
b - elementul curent
a – elementul precedent
max – elementul de valoare maxima, varful local precedent

pozmax – pozitia ultimului varf local
n – numarul curent de elemente citite din fisier

Primul element din sir este varf local. Vom retine pentru acest var valoarea lui (max) si pozitia lui (pozmax).
Vom citi elementele urmatoare din fisier, de fiecare data vom avea grija sa retinem si valoarea elementului precedent.
Daca intalnim o valoare maxim in fisier inseamna ca acesta este noul varf local, il afisam si ii vom pastra pozitia (pozmax) si valoarea (max).
Daca elementul precedent este varf local (elementul precedet este elementul n-1 citit din fisier) il vom afisa si vom reintializa pozitia ultimului varf local intalnit.

Programul se bazeaza pe o citire secventiala a numerelor din fisier si foloseste o singura instructiune repetitiva. Prin urmare, timpul lui de executie este mic.
Programul foloseste un numar mic de variabile si nu foloseste tablouri cu numar variabil de elemente. Prin urmare, spatiul rezervat in memorie pentru variabile, la executie, este mic.