

SUBIECTUL I

1. Variabila întregă n memorează un număr natural cu exact 4 cifre. Care din expresiile C++ de mai jos este echivalentă cu cea alăturată? **$n/100\%10\%2==0$**

- a. $n\%100/10\%2!=1$ b. $n\%1000\%2==0$ c. **$n/100\%2==0$** d. $n/10\%10!=1$

2. Se consideră algoritmul alăturat, descris în pseudocod.

citește n

(număr natural, $n > 1$)

ok ← 0

cât timp $n > 0$ execută

$c \leftarrow n\%10$

dacă $c > 5$ și $c\%2 = 0$ atunci

 ok ← 1

 altfel

 ok ← 0

sf. dacă

dacă ok = 1 atunci

 scrie c , ‘ ‘

 ok ← 1

sf. dacă

$n \leftarrow [n/10]$

sf. cât timp

dacă ok = 0 atunci

 scrie "nu"

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y , iar cu $[z]$ partea întregă a numărului real z .

- a) Scrieți ce se afișează dacă numărul citit este $n=4576$ Răspuns 6
 b) Scrieți cea mai mare valoare cu exact 3 cifre care poate fi citită pentru n astfel încât să se afișeze, în această ordine, numerele 8, 6. Răspuns 968
 c) Scrieți programul C++ corespunzător algoritmului dat.

```
#include <iostream>
using namespace std;
int n,ok,ok1,c;
int main()
{
    cout << "n="; cin>>n;
    ok=0;
    while(n>0) {
        c=n%10;
        if ((c>5) && (c%2==0))
            ok1=1;
        else ok1=0;
        if (ok1==1) {
            cout<<c<<" ";
            ok=1; }
        n=n/10;
    }
```

```

}
if (ok==0) cout<<"nu";
return 0;
}

```

d) Scrieți în pseudocod un algoritm echivalent cu cel dat care să utilizeze o structură repetitivă de alt tip în locul structurii **cât timp ... execută**.

```

citește n
(număr natural, n>1)
ok<-0
repetă
  c<-n%10
  dacă c>5 și c%2=0 atunci
    ok1<-1
  altfel
    ok1<-0
sf. dacă
dacă ok=1 atunci
  scrie c, ' '
  ok<-1
sf. dacă
n<-[n/10]
până când n=0
dacă ok=0 atunci
  scrie "nu"

```

SUBIECTUL II

- Se consideră un graf neorientat cu 6 noduri și 9 muchii. Numărul de muchii ce trebuie adăugate, pentru ca graful obținut să fie complet, este:
 - 5
 - 6**
 - 12
 - 25
- Se consideră secvența de instrucțiuni alăturată, în care variabilele i și j sunt de tip întreg, iar variabila s poate memora un șir de cel mult 20 de caractere. Șirul afișat în urma executării secvenței este:
 - aureatbacal
 - bacalaureab
 - taealaureat**
 - taerualacab

```

1 #include <iostream>
2 #include <cstring>
3 int i, j;
4 char s[20];
5 using namespace std;
6 int main()
7 {
8     strcpy(s, "bacalaureat");
9     j=strlen(s);
10    for (i=0; i<3; i++)
11        if (s[i]!=s[j-i-1])
12            s[i]=s[j-i-1];
13    cout<<s;
14    return 0;
15 }

```

Output window: taealaureat, Process returned 0 (0x0), execution time : 0.470 s, Press any key to continue.

- Se consideră un arbore cu rădăcină având 6 noduri, numerotate de la 1 la 6, și muchiile [1,2], [1,3], [2,4], [3,5], [3,6]. Considerând că rădăcina este nodul numerotat cu 1 și că ea este situată pe nivelul 0 al arborelui, scrieți toate nodurile situate pe nivelul 2 al arborelui dat.

Răspuns: 4, 5, 6.

4. Variabilele e1 și e2, declarate alăturat, memorează numele și data nașterii pentru câte un elev. Știind că cei doi elevi sunt născuți în ani diferiți, scrieți secvența de instrucțiuni care afișează pe ecran numele elevului de vârstă mai mare.

```

if (e1.d.an<e2.d.an)
    cout<<e1.nume;
else cout<<e2.nume;

```

5. Scrieți un program C++ care citește de la tastatură un număr natural n ($2 \leq n \leq 20$), și construiește în memorie un tablou bidimensional cu n linii și n coloane, având proprietățile:
- Toate elementele situate pe diagonala principală sunt nule;
 - Fiecare linie conține, începând de la diagonala principală în jos, de la dreapta la stânga un șir crescător de numere consecutive, iar începând cu diagonala principală în sus, de la stânga la dreapta, tot un șir crescător de numere consecutive.

```

1 #include <iostream>
2 int n,i,j,k,a[20][20];
3 using namespace std;
4 int main()
5 { cout << "n=";cin>>n;
6   for (i=1;i<=n;i++)
7     for (j=1;j<=n;j++)
8       if (i==j) a[i][j]=0;
9   for (i=1;i<=n;i++)
10    {k=1;
11     for (j=1;j<=n;j++)
12       if (i<j) {
13         a[i][j]=k;
14         k++; }
15     for (i=n;i>=1;i--)
16     {k=1;
17      for (j=n;j>=1;j--)
18        if (i>j){
19          a[i][j]=k;
20          k++; } }
21     for (i=1;i<=n;i++)
22     {for (j=1;j<=n;j++)
23      cout<<a[i][j]<<' ';
24      cout<<endl;}
25   return 0;
26 }
27

```

Execution output:

```

n=5
0 1 2 3 4
1 0 1 2 3
2 1 0 1 2
3 2 1 0 1
4 3 2 1 0
Process returned 0 (0x0)   execution time : 3.407 s
Press any key to continue.

```

SUBIECTUL III

1. Utilizând metoda backtracking se generează în ordine crescătoare numerele naturale de câte patru cifre din mulțimea $A=\{1, 2, 3, 4, 5\}$, numere care nu conțin două cifre impare alăturate. Primele opt numere generate sunt, în această ordine: 1212, 1214, 1221, 1222, 1223, 1224, 1225, 1232. Numărul de valori generate care au cifra miilor egală cu 2 și cifra unităților egală cu 4 este:
- a. 20 **b. 16** c. 12 d. 9
2. Scrieți ce se afișează în urma apelului de mai jos pentru subprogramul f, definit alăturat. f(12345);

```

1 #include <iostream>
2 using namespace std;
3 void f(int x)
4 {
5   cout<<"*";
6   if (x>0)
7   {
8     cout<<x;
9     f(x/100);
10    cout<<"*";
11  }
12 }
13 int main()
14 {
15   f(12345);
16   return 0;
17 }
18

```

Execution output:

```

*12345*123*1*****
Process returned 0 (0x0)
execution time : 0.401 s
Press any key to continue.

```

3. Subprogramul inter are patru parametri:
- na, nb prin care primește câte un număr natural ($0 < na < 100$, $0 < nb < 100$);

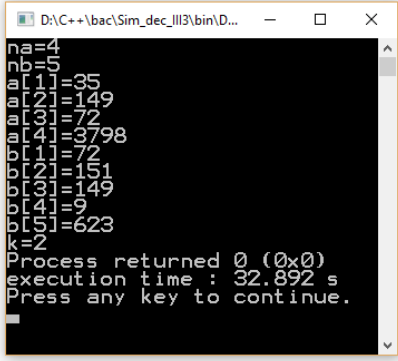
- a, b prin care primește câte un tablou unidimensional care memorează câte o mulțime de na, respectiv nb numere naturale, fiecare având cel mult patru cifre;

Subprogramul returnează numărul de elemente aparținând intersecției celor două mulțimi menționate. Scrieți în C++ definiția completă a subprogramului inter.

```

1  #include <iostream>
2  int na, nb, a[100], b[100];
3  using namespace std;
4  int inter(int na, int nb, int a[100], int b[100])
5  {int i,j,k=0;
6    for(i=1; i<=na; i++)
7    {
8      cout<<"a["<<i<<"]=""; cin>>a[i];
9    }
10   for(j=1; j<=nb; j++)
11   {
12     cout<<"b["<<j<<"]=""; cin>>b[j];
13   }
14   for (i=1;i<=na;i++)
15     for (j=1;j<=nb;j++)
16       if (a[i]==b[j]) k++;
17   cout<<"k="<<k;
18 }
19 int main()
20 {
21   cout << "na="; cin>>na;
22   cout << "nb="; cin>>nb;
23   inter(na,nb,a,b);
24   return 0;
25 }
26

```



4. Fișierul bac.txt conține un șir de cel puțin 11 și cel mult un milion de numere naturale despărțite prin câte un spațiu. Fiecare număr are cel puțin două și cel mult nouă cifre. Primul termen al șirului are numărul de ordine 1, al doilea are numărul de ordine 2, etc.

Se citește șirul din fișier și se cere ca, utilizând un algoritm eficient din punct de vedere al timpului de execuție, să se determine și să se afișeze pe ecran numărul de ordine al următorului termen al șirului care este precedat în fișier de un număr maxim de valori care au cifra zecilor egală cu a sa. Dacă sunt mai mulți termeni cu această proprietate se afișează numărul de ordine doar al unuia dintre ei.

```

main.cpp x bac.txt
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  ifstream f("bac.txt");
5  int i,k,x,y,ok,p;
6  int main()
7  {
8    k=0;i=0;ok=0;
9    f>>x;i++;
10   while(f>>y)
11   {
12     i++;
13     if (x/10%10==y/10%10)
14       {k++;
15        x=y;
16        ok=1;}
17     else ok=0;
18     if (ok==1) p=i;
19   }
20   cout<<p;
21   return 0;
22 }
23

```

