

## Subiect Informatica - 2023

- I.    1.d  
      2.a  
      3.c  
      4.b  
      5.b

- II.   1. a) 975  
      b) 875, 965  
      c) 

```
#include <iostream>
using namespace std;
int main() {
    int x,p=1,m=-1,c;
    cin>>x;
    while (p<=x) {
        c=(x/p)%10;
        if (c>m) {
            m=c;
            p=p*10;
        }
        else {
            x=(x/(p*10))*p+x%p;
        }
    }
    if (m>=0) {
        cout<<x;
    }
    else {
        cout<<"nul";
    }
}
```
- d) citeste x  
   p=1; m=-1  
   └daca p<=x atunci  
   | └executa  
   || c=[x/p]%10

```

|| Țdaca c>m atunci
||| m=c; p=p*10
||| altfel
||| x=[x/(p*10)]*p+x%p
||  L■
| |-cat timp p<=x
L■

```

```

Țdaca m>=0 atunci scrie x
| altfel scrie "nul"
L■

```

2. (mare,ocean)  
(ocean,ploaie,râu)

```

3. struct flori {
    char nume[20];
    struct {
        int numarFire,pretFir;
    }stoc;
}f[10];

```

III. 1. int abundant(int n) {  
 int sdivn=n;  
 for (int i=1; i<=n/2; i++){  
 if (n%i==0)  
 {  
 sdivn+=i;  
 }  
 }  
 for (int i=1; i<n; i++) {  
 int sdiv=i;  
 for (int j=1; j<=i/2; j++){  
 if (i%j==0){  
 sdiv+=j;  
 }  
 }  
 }  
}

```

        if (sdiv/i>=sdivn/n){
            return 0;
        }
    }
    return 1;
}

```

2. #include <iostream>

```

using namespace std;
int main() {
    int ns,np,a[1001][51],i,j;
    cin>>ns>>np;
    for (i=1; i<=ns; i++){
        for (j=1; j<=np; j++){
            cin>>a[i][j];
        }
    }
    int max,maxj;
    for (i=1; i<=ns; i++) {
        max=-1;
        for (j=1; j<=np; j++) {
            if (a[i][j]>max) {
                max=a[i][j];
                maxj=j;
            }
        }
        cout<<i<<": "<<maxj<<" ";
    }
    return 0;
}

```

3. b)

```

#include <iostream>
#include <fstream>
using namespace std;
int main() {
    int x,suffix[100]={0},prefix[100]={0},suf,pref;

```

```

while (fin>>x) {
    if (x>=100) {
        suf=x%100;
        pref=x;

        while (pref >= 100) {
            pref = pref/10;
        }
        suffix[suf]++;
        prefix[pref]++;
    }
}
int cnt=0;
for (int i=10; i<=99; i++) {
    if (prefix[i] == suffix[i] && suffix[i]!=0)
    {
        cnt++;
    }
}
cout<<cnt;
}

```

- a) Pentru a rezolva aceasta problem vom folosi 2 vectori de frecventa de max. 100 elemente pentru a contoriza aparitiile fiecarui sufix si prefix. Sufixul este reprezentat de ultimele 2 cifre din fiecare numar, iar prefixul este reprezentat de primele 2 cifre, calculat cu ajutorul unui while. Se parcurge vectorul de frecventa si se verifica daca numarul de aparitii al unui prefix este egal cu nr. de aparitii al unui sufix reprezentat de acelasi numar, dar si daca este diferit de 0.

Algoritmul este eficient din punct de vedere al memoriei, deoarece folosim tot ceea ce declaram. De exemplu daca am fi stocat cifrele intr-un vector, exista posibilitatea ca o mare parte din acesta sa nu fie folosita. De asemenea, algoritmul are o complexitate in timp liniara  $O(N)$  – unde  $N$  este numarul de elemente din fisierul bac.txt.

Greseli:

I.2 -0.4p

5 -0.4p

II.1.a) -0.6p

II.2. -0.3p

3. -0.3p (denumire struct gresita, 2 denumiri de variabile gresite)

Total : -2p