Universitatea de Stat din Tiraspol

Facultatea Fizică, Matematică și Tehnologii Informaționale

Catedra Informatică și Tehnologii Informaționale

Programele de studiu: *Matematică și Informatică*, *Informatică*, *Fizică și Informatică*

Grupa academică: \_\_\_\_\_\_\_2MI\_\_\_\_\_\_\_\_\_\_\_

Numele și prenume studentului: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Razloga Anastasia\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Evaluarea 1

la cursul *Structuri de date și algoritmi*

1. Fie **a** un număr întreg citit de la tastatură. Care va fi valoarea expresiei **a % 4** pentru **a = 1234**?

*Răspuns*:

#include <iostream>

using namespace std;

int main(){

int a ,mod;

cin>>a;

mod=a%4;

cout<<mod;

}

Raspuns: 2

2. Scrieți o funcție cu parametrul **a**, unde **a** este număr natural mai mic decât 30 000, care va returna suma cifrelor lui **a**.

#include <iostream>

using namespace std;

suma(int i){

int m;

while (i != 0)

{

m = m + i % 10;

i = i / 10;

}

cout << "Suma cifrelor lui n este: " << m;

return 0;

}

int main(){

int a ;

cout<<"a<30 000 :"<<endl;

cin>>a;

suma(a);

}

Raspuns :

23458

suma = 22

3. Scrieți doi algortimi care ordonează crescător un masiv de numere întregi prin două metode diferite.

*Răspuns*:

Algoritmul buble:

c=0;

while (c==0)

{

c=1;

for ( i = 1; i <= n-1 ; i++ )

if ( v[i] > v[i+1] )

{ aux = v[i];

v[ji+1] = aux;

aux=0;

}}

Algoritmul de implementare:

for(i=0;i<=n-2;i++){

for(j=i+1;j<=n-1;j++){

if(v[i]>v[j]){

aux=v[i];

v[i]=v[j];

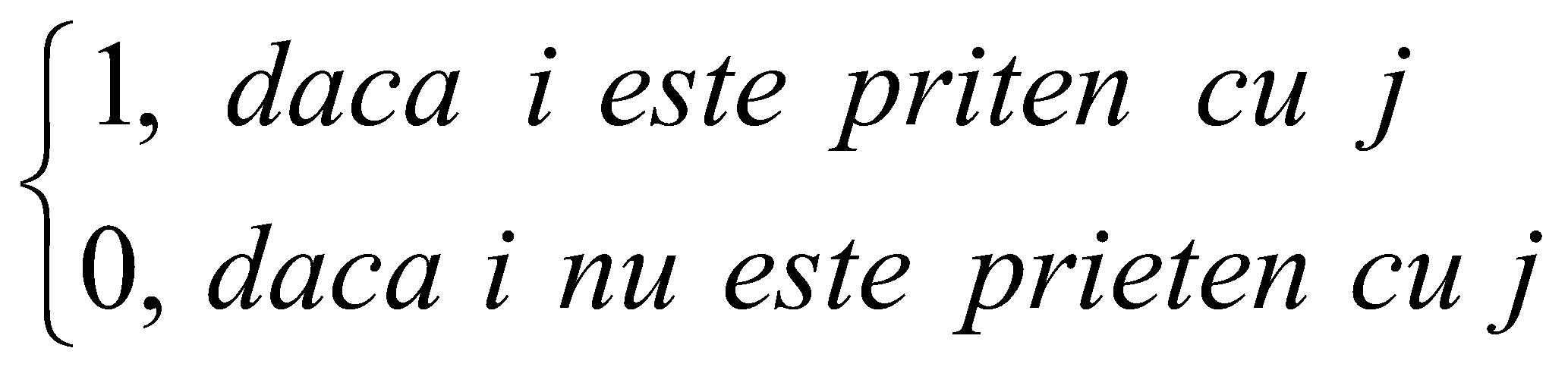
v[j]=aux;

}

}

}

4. Considerăm un grup de *n* persoane. Matricea *A*(*n*, *n*) este definită astfel:

*a*(*i*, *j*) = , *a*(*i*, *i*) = 1.

Scrieţi un algoritm care determină prietenii comuni din grup ai persoanelor *p*, *q*, unde *p* şi *q* sînt numere naturale nenule date mai mici sau egale cu *n*.

*Răspuns*:

#include <iostream>

using namespace std;

int main(){

int n,i,j,a[100][100],p,q,c,v[100];

cout<<"Dati dimensiunile matricii (n x n)"<<endl;

cin>>n;

for(i=0;i<n;i++)

for(j=0;j<n;j++){

cout<<"a["<<i+1<<","<<j+1<<"]= ";

cin>>a[i][j];

}

cout<<"Matricea este : "<<endl;

for(i=0;i<n;i++){

for(j=0;j<n;j++)

cout<<a[i][j]<<" ";

cout<<endl;

}

cout<<"Introduceti p : ";

cin >> p ;

cout<<"Introduceti q : ";

cin>> q;

for (j = 0;j < n; j++){

if ( (a[p][j] == 1) && (a[q][j] == 1) ) {

v[c] = j;

c++;

}

}

cout << "Prietenii comuni:";

for (i = 0;i < c; i++)

cout << v[i]+1 << ' ';

return 0;

}

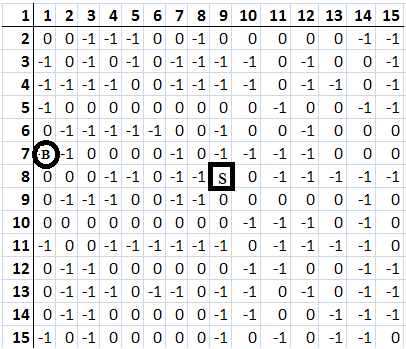
5. O structură de date de tip stivă a fost „umplută” cu următoarele elemente în această ordine:

10, –23, 8, 4, 29, 7, –66, 43, unde 43 este ultimul element pus în *stivă*.

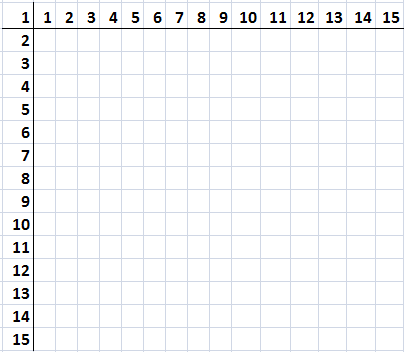
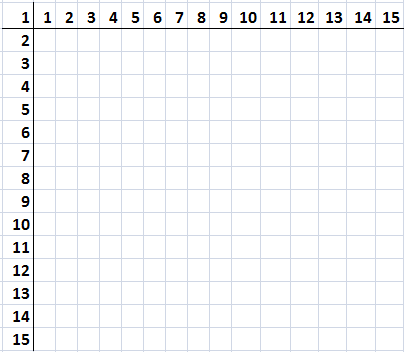
Care va fi al treilea element scos din *stivă*?

*Răspuns*:7

6. Construiți soluția pentru următoarea problemă a Labirintului:



*Rezolvare*:

 🡺 

*Răspuns*:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | -1 | -1 | -1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | -1 | -1 |
| -1 | 0 | -1 | 0 | -1 | 0 | -1 | -1 | -1 | -1 | 0 | -1 | 0 | 0 | -1 |
| -1 | -1 | -1 | -1 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | -1 | -1 | 0 | -1 |
| -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | -1 |
| 0 | -1 | -1 | -1 | -1 | -1 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | 0 | 0 |
| **14** | -1 | 12 | **11** | **10** | **9** | -1 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 |
| **13** | **14** | **13** | -1 | -1 | **8** | -1 | -1 | **1** | 0 | -1 | -1 | -1 | -1 | -1 |
| **12** | -1 | -1 | -1 | **8** | **7** | -1 | -1 | **2** | 0 | 0 | 0 | 0 | -1 | 0 |
| **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | -1 | -1 | -1 | 0 | -1 | 0 |
| -1 | **11** | **10** | -1 | -1 | -1 | -1 | -1 | -1 | 0 | -1 | -1 | -1 | -1 | 0 |
| 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 |
| 0 | -1 | -1 | -1 | 0 | -1 | -1 | 0 | -1 | -1 | 0 | -1 | 0 | -1 | -1 |
| 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 | -1 |
| -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 0 | -1 | -1 | 0 |

Drumul:

[8][9]

[9][9]

[10][9]

[10][8]

[10][7]

[10][6]

[10][5]

[10][4]

[10][3]

[10][2]

[10][1]

[9][1]

[8][1]

[7][1]