**Lucrare de laborator Nr.1**

Problema 1. Se dau numerele întregi a, b, c, d, e, diferite fiecare două. Scrieți un algoritm care determină numărul al 2-lea (numărat din stânga) la aranjarea numerelor în ordine crescătoare.

**Rezolvare:**

#include <iostream>

using namespace std;

int main()

{

int n,a[1000],i,j,aux;

cin>>n;

for(i=1;i<=n;i++) cin>>a[i];

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

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

if(a[j]<a[i]){

aux=a[i];

a[i]=a[j];

a[j]=aux;}

cout<<"Vectorul sortat crescator este: ";

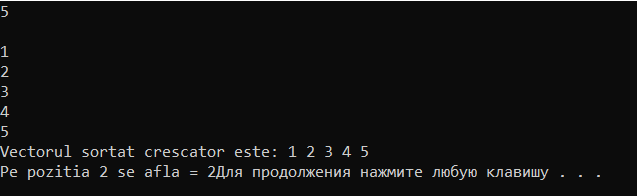
for(i=1;i<=n;i++)

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

cout<<endl<<"Pe pozitia 2 se afla = "<<a[2]<<"";

return 0;

}



Problema 2. Se dă numărul natural n, < n < . Scrieți un algoritm care determină cifra zecilor.

**Rezolvare:**

#include <iostream>

using namespace std;

int main(){int n,c,k;

k=0;

cout<<"100 < n < 1000000 Dati n= "; cin >> n;

while(k<2){

c = n%10;

n = n / 10;

k++;

}

cout<<"Cifra zecilor este = "<< c;

}



36. Se dă un număr natural n mai mic decât 27. Să se afișeze litera a cărui nr de ordine în alfabetul latin este n.

#include <iostream>

#include<cstring>

using namespace std;

main()

{

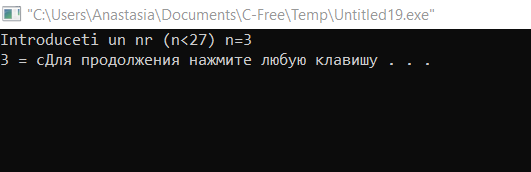
int n;

cout<<"Introduceti un nr (n<27) n=";

cin>>n;

char a = (char) n +96; cout<<n<<" = "<<a;

}



37. Se dă o literă a alfabetului latin. Să se afișeze numărul de ordineal ei în acest alfabet. De exemplu, pentru litera M se va afișa 13.

#include <iostream>

#include<cstring>

using namespace std;

main()

{

char b;

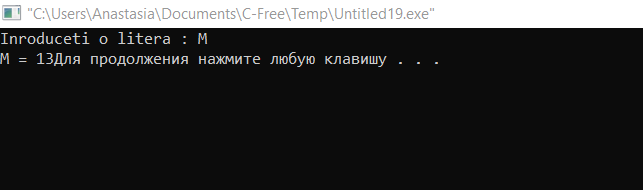
cout<<"Inroduceti o litera : ";

b= getchar();cout<<b;

b=tolower(b);

int a = b - 96; cout<<" = "<<a;

}



**Lucrare de laborator Nr.2 (Varianta V)**

Problema 1. **Se dă un număr natural n, < n <. Scrieți un algoritm care determină divizorii numărului n, care se încep cu cifră impară.**

**#include <iostream>**

**using namespace std;**

**void divimp(int x)**

**{**

**for(int i = 1; i <= x / 2; i++)**

**{**

**if(x % i == 0){**

**if(i%2!=0)**

**cout << i << " ";**

**}}**

**}**

**int main()**

**{**

**int n;**

**cout << "(10^2 < N < 10^6) N = ";**

**cin >> n;**

**divimp (n);**

**return 0;**

**}**

Problema 2. **Se dă șirul R, care reprezintă scrierea romană a unui număr necunoscut n. Scrieți un algoritm care afișează n.**

**#include <iostream>**

**using namespace std;**

**char n[10];**

**int out[10];**

**int i;**

**void convert()**

**{**

**while(n[i]!=NULL)**

**{**

**if(n[i]== 'I')**

**out[i]=1;**

**if(n[i]== 'V')**

**out[i]=5;**

**if(n[i]== 'X')**

**out[i]=10;**

**if(n[i]== 'L')**

**out[i]=50;**

**if(n[i]== 'C')**

**out[i]=100;**

**if(n[i]== 'D')**

**out[i]=500;**

**if(n[i]== 'M')**

**out[i]=1000;**

**i++;**

**}**

**}**

**int ar(int f)**

**{**

**int m = 0,t=0,g=0;**

**i=0;**

**cout<<"Introduceti un nr Roman (I,V,X...): ";**

**cin>>n;**

**convert();**

**while(out[m]!=NULL)**

**{**

**if(out[m]>=out[m+1])**

**{**

**t=t+out[m];**

**}**

**else**

**if(out[m]<out[m+1])**

**{**

**g=out[m+1]-out[m];**

**m++;**

**}**

**f=t+g;**

**m++;**

**}**

**return f;**

**}**

**int main (void)**

**{**

**int g;**

**cout<<endl;**

**g = ar(g);**

**cout<<"Numarul necunoscut este : "<<g;**

**}**

**Lucrare de laborator Nr.3**

Problema 7. 

#include <iostream>

using namespace std;

int v[100], n,t,j;

int main() {

int i;

cout<<"Nr de elemente al vectorului n>100 n=";

cin >> n;

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

cin >> v[i];

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

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

cout<<endl<<"Introduceti t 1<t<n t= ";

cin >> t;

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

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

int aux = v[i];

v[i] = v[j];

v[j] = aux;

}

}

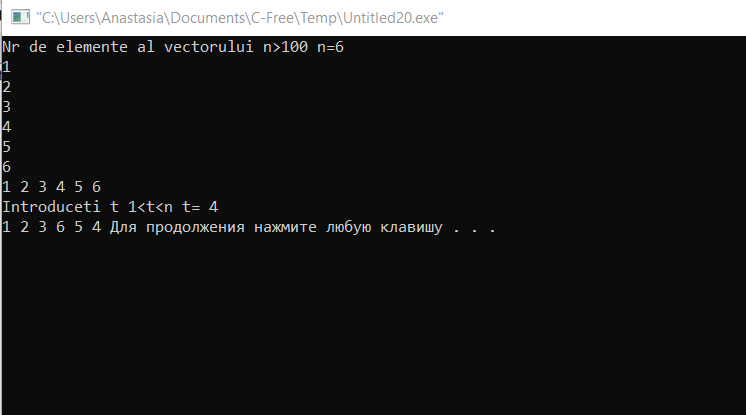
for(i=1;i<t;i++)

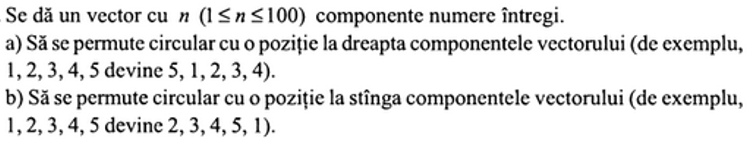
cout<<v[i]<<" ";

for(i=n;i>=t;i--)

cout<<v[i]<<" ";

}



Problema 27. 

**a)** #include <iostream>

using namespace std;

int v[100], n,x;

int main() {

int i;

cout<<"Nr de elemente al vectorului n>100 n=";

cin >> n;

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

cin >> v[i];

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

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

cout<<endl;

x=v[n];

for(i=n-1;i>=1;i--)

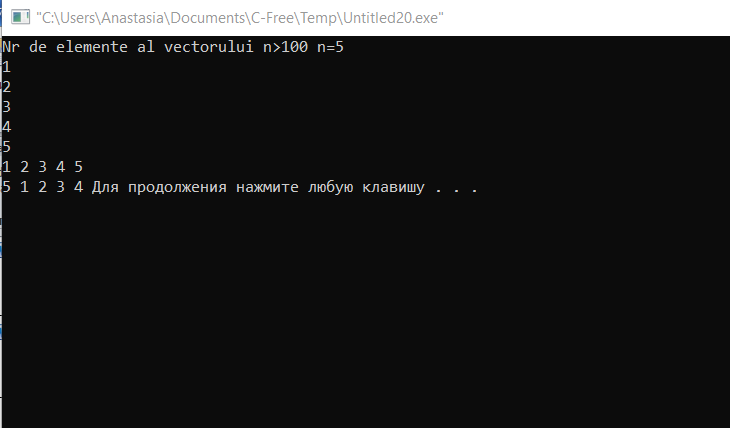
v[i+1]=v[i];

v[1]=x;

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

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

}



**b)** #include <iostream>

using namespace std;

int v[100], n,x;

int main() {

int i;

cout<<"Nr de elemente al vectorului n>100 n=";

cin >> n;

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

cin >> v[i];

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

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

cout<<endl;

x=v[1];

for(i=2;i<=n;i++)

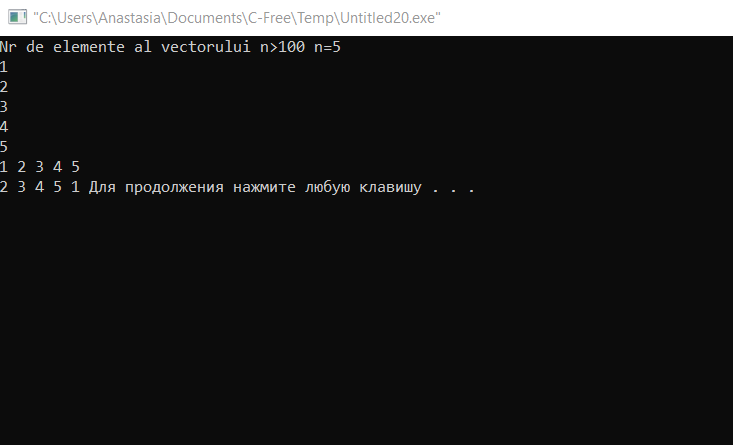
v[i-1]=v[i];

v[n]=x;

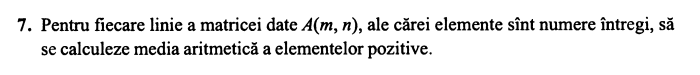
for (i = 1; i < n+1; i++)

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

}



**Lucrare de laborator Nr.4**



#include<iostream>

using namespace std;

main(){

int a[20][30], i, j, m, n, s=0,c;

cout<<"Dati numarul de linii m=";

cin>>m;

cout<<"Dati numarul de coloane n=";

cin>>n;

for(i=0; i<=m-1; i++)

for(j=0; j<=n-1; j++){

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

cin>>a[i][j];

if(a[i][j]>0)

s+=a[i][j];

}

for(i=0; i<=m-1; i++)

for(j=0; j<=n-1; j++){

c++;

}

cout<<"Matricea introdusa: "<<endl;

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

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

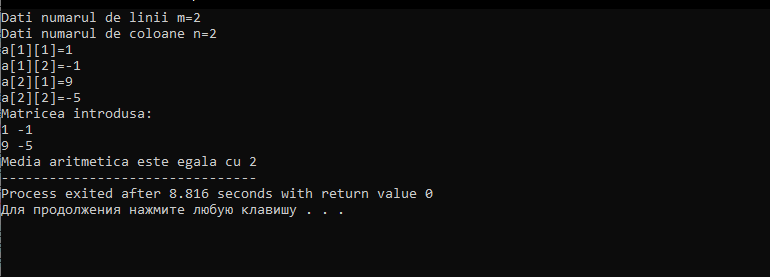
cout<<a[i][j]<<" ";

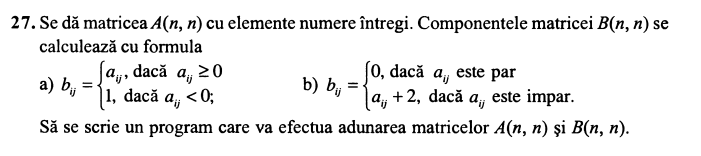
cout<<endl;

}

cout<<"Media aritmetica este egala cu "<< (float) s/c;

}





#include <iostream>

using namespace std;

main(){

int n,i,j,a[30][30],b[30][30],c[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii A sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if (a[i][j]>=0){b[i][j]=a[i][j];

}else if (a[i][j]<0){

b[i][j]=1; }}

cout<<"Elementele matricii B sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

cout<<"A + B = "<<endl;

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

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

c[i][j]=a[i][j]+b[i][j];

}}

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

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

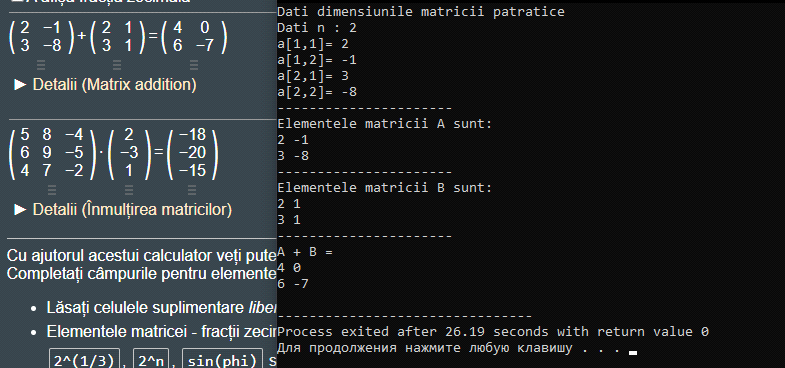
cout<<c[i][j]<<" ";

cout<<endl;

}

return 0;

}



b)

#include <iostream>

using namespace std;

main(){

int n,i,j,a[30][30],b[30][30],c[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii A sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if (a[i][j]%2==0){b[i][j]=0;

}else if (a[i][j]%2!=0){

b[i][j]=a[i][j]+2; }}

cout<<"Elementele matricii B sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

cout<<"A + B = "<<endl;

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

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

c[i][j]=a[i][j]+b[i][j];

}}

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

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

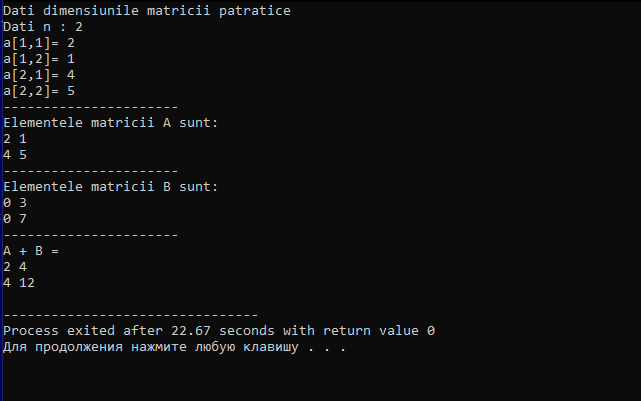
cout<<c[i][j]<<" ";

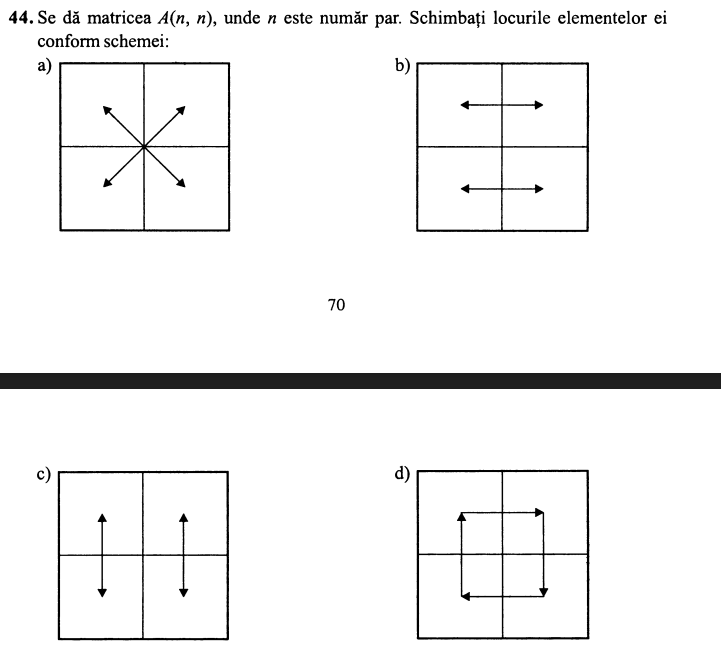
cout<<endl;

}

return 0;

}





a)

#include <iostream>

using namespace std;

int main(){

int n,i,j,a[30][30],k,v[30],t[30],b[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if ((i < n/2 )&&(j < n/2)){

b[i + (n/2)][j + (n/2)] = a[i][j];

}

else if ((i >= n/2 )&& (j >= n/2)){

b[i - int(n/2)][j - int(n/2)] = a[i][j];

}

else if (i >= n/2 >= j){

b[i - int(n/2)][j + int(n/2)] = a[i][j];

}

else if (i <= n/2 <= j){

b[i + int(n/2)][j - int(n/2)] = a[i][j] ;

}

cout<<"Elementele matricii obtinute in varianta a) sunt: "<<endl;

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

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

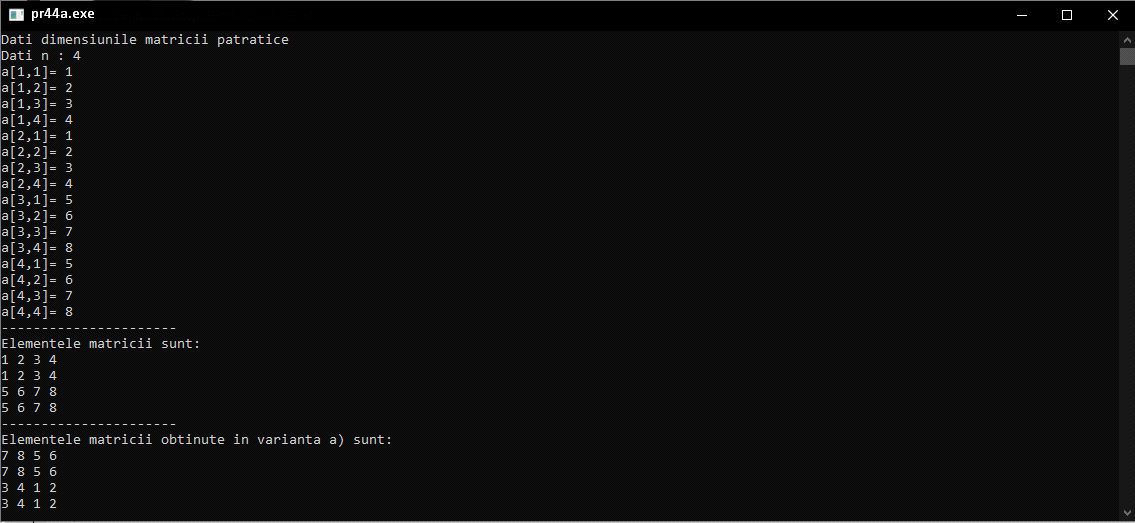
cout<<b[i][j]<<" ";

cout<<endl;

}

return 0;

}



b)

#include <iostream>

using namespace std;

int main(){

int n,i,j,a[30][30],k,v[30],t[30],b[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if ((i < n/2 )&&(j < n/2)){

b[i][j] = a[i ][j + (n/2)];//cadranul 1 cu cadranul2

}

else if ((i >= n/2 )&& (j >= n/2)){

b[i-(n/2)][j] = a[i-(n/2)][j-(n/2)] ; //cadranul 2 cu cadranul 1

}

else if (i >= n/2 >= j){

b[i][j] = a[i][j + (n/2)] ;//cadranul 3 cu cadranul 4

}

else if ((i <= n/2 <= j)){

b[i+(n/2)][j] = a[i+(n/2)][j-(n/2)] ;//cadranul 4 cu cadranul 3

}

cout<<"Elementele matricii obtinute in varianta b) sunt: "<<endl;

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

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

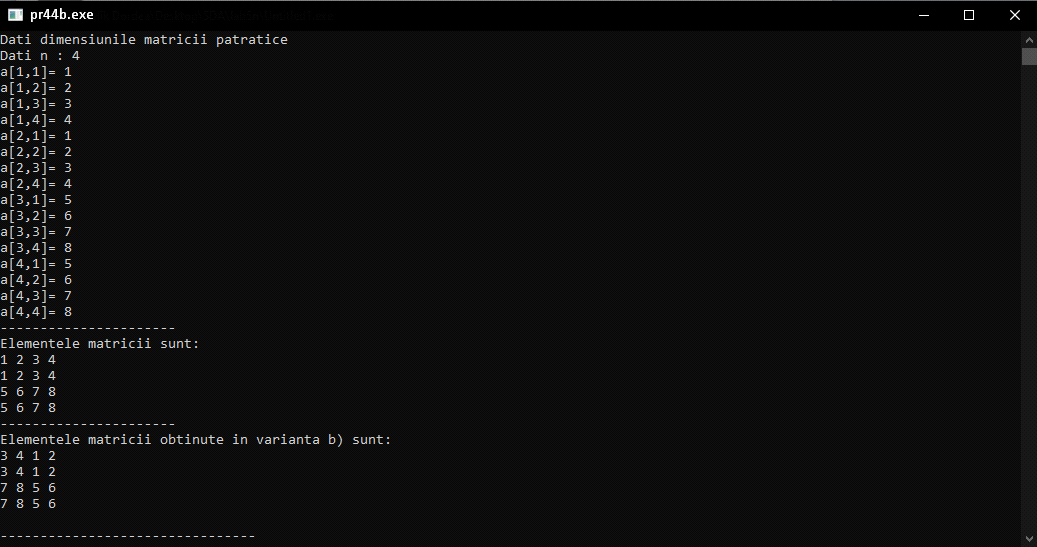
cout<<b[i][j]<<" ";

cout<<endl;

}

return 0;

}



c)

#include <iostream>

using namespace std;

int main(){

int n,i,j,a[30][30],k,v[30],t[30],b[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if ((i < n/2 )&&(j < n/2)){

b[i][j] = a[i+ (n/2) ][j];//cadranul 1 cu cadranul 3

}

else if (i >= n/2 >= j){

b[i][j] = a[i-(n/2)][j] ;//cadranul 3 cu cadranul 1

}

else if ((i >= n/2 )&& (j >= n/2)){

b[i-(n/2)][j] = a[i][j] ; //cadranul 2 cu cadranul 4

}

else if ((i <= n/2 <= j)){

b[i+(n/2)][j] = a[i][j] ;//cadranul 4 cu cadranul 3

}

cout<<"Elementele matricii obtinute in varianta c) sunt: "<<endl;

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

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

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

cout<<endl;

}

return 0;

}



d)

#include <iostream>

using namespace std;

int main(){

int n,i,j,a[30][30],k,v[30],t[30],b[30][30];

cout<<"Dati dimensiunile matricii patratice"<<endl;

cout<<"Dati n : ";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<<"----------------------"<<endl;

cout<<"Elementele matricii sunt: "<<endl;

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

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

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

cout<<endl;

}

cout<<"----------------------"<<endl;

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

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

if ((i >= n/2 )&& (j >= n/2)){

b[i-(n/2)][j] = a[i-(n/2)][j-(n/2)] ; //cadranul 2 cu cadranul 1

}

else if ((i <= n/2 <= j)){

b[i+(n/2)][j] = a[i][j] ;//cadranul 4 cu cadranul 2

}

if ((i < n/2 )&&(j < n/2)){

b[i+(n/2)][j] = a[i+(n/2)][j+(n/2)];//cadranul 3 cu cadranul 4

}

else if (i >= n/2 >= j){

b[i-(n/2)][j] = a[i][j] ;//cadranul 1 cu cadranul 3

}}

cout<<"Elementele matricii obtinute in varianta c) sunt: "<<endl;

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

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

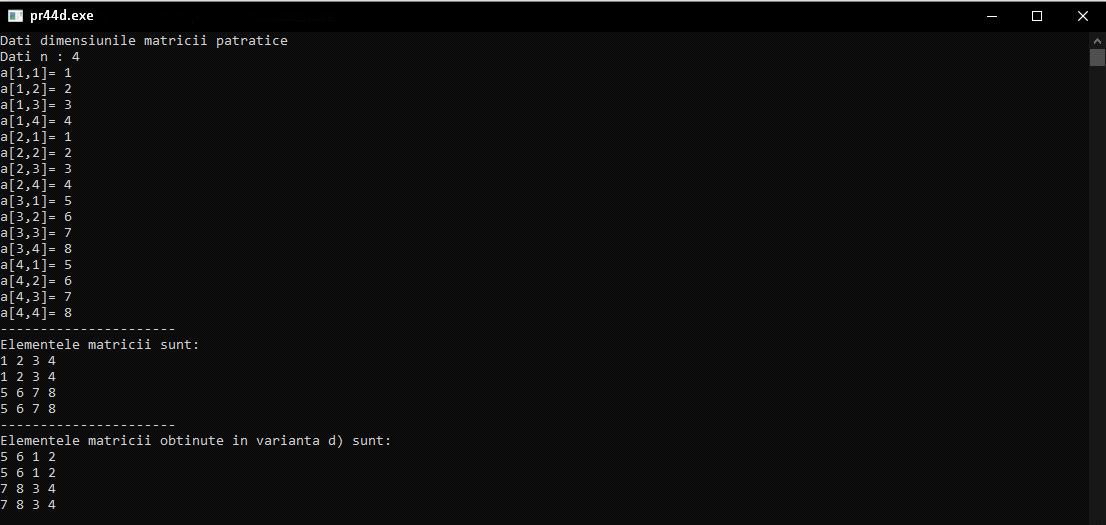
cout<<b[i][j]<<" ";

cout<<endl;

}

return 0;

}



**Lucrare de laborator Nr.5**

Problema 8. **Scrieți un algoritm care rezolvă problema labirintului.**

#include <iostream>

#include <stdlib.h>

using namespace std;

struct tpunct {

int x,y;

};

int a[50][50];

tpunct stiva[100];

tpunct coada[100];

int i,j,m,n,sr,sc,br,bc,s;

int vc, vs;

int sfc, k;

bool f;

int main() {

cout << "Introdu dimensiunile labirintului: \n";

cout << "m = "; cin >> m;

cout << "n = "; cin >> n;

for (i = 1; i < m; i++)

for (j = 1; j < m; j++)

a[i][j] = rand() % 2 - 1;

cout << "Matricea initiala: \n";

for (i = 1; i < m; i++) {

for (j = 1; j < n; j++)

if (a[i][j] < 0)

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

else

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

cout << '\n';

}

cout << "Introdu coordonatele soarecelui: ";

cout << "\nsr = "; cin >> sr;

cout << "sc = "; cin >> sc;

a[sr][sc] = 1;

i = sr;

j = sc;

cout << "Introdu coordonatele brinzei: ";

cout << "\nbr = "; cin >> br;

cout << "bc = "; cin >> bc;

vc = 1;

sfc = 1;

coada[vc].x = sr;

coada[vc].y = sc;

do {

f = false;

if ((i - 1 > 0) && (a[i - 1][j] == 0)) {

a[i - 1][j] = a[i][j] + 1;

sfc = sfc + 1;

coada[sfc].x = i - 1;

coada[sfc].y = j;

f = true;

}

if ((i + 1 <= m) && (a[i + 1][j] == 0)) {

a[i + 1][j] = a[i][j] + 1;

sfc = sfc + 1;

coada[sfc].x = i + 1;

coada[sfc].y = j;

f = true;

}

if ((j - 1 > 0) && (a[i][j - 1] == 0)) {

a[i][j - 1] = a[i][j] + 1;

sfc = sfc + 1;

coada[sfc].x = i;

coada[sfc].y = j - 1;

f = true;

}

if ((j + 1 <= n) && (a[i][j + 1] == 0)) {

a[i][j + 1] = a[i][j] + 1;

sfc = sfc + 1;

coada[sfc].x = i;

coada[sfc].y = j + 1;

f = true;

}

if (f == false) {

vc = vc + 1;

i = coada[vc].x;

j = coada[vc].y;

}

} while ((coada[vc].x != br) || (coada[vc].y != bc));

cout << "Matricea nr. 2: \n";

for (i = 1; i < m; i++) {

for (j = 1; j < n; j++)

if (a[i][j] < 0)

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

else

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

cout << '\n';

}

k = a[br][bc];

cout << "k =" << k << '\n';

vs = 1;

stiva[vs].x = br;

stiva[vs].y = bc;

i = br;

j = bc;

do {

f = false;

if ((i - 1 > 0) && (a[i - 1][j] == k - 1)) {

vs = vs + 1;

stiva[vs].x = i - 1;

stiva[vs].y = j;

f = true;

}

if ((i + 1 <= m) && (a[i + 1][j] == k - 1) && (f == false)) {

vs = vs + 1;

stiva[vs].x = i + 1;

stiva[vs].y = j;

f = true;

}

if ((j + 1 <= n) && (a[i][j + 1] == k - 1) && (f == false)) {

vs = vs + 1;

stiva[vs].x = i;

stiva[vs].y = j + 1;

f = true;

}

if ((j - 1 > 0) && (a[i][j - 1] == k - 1) && (f == false)) {

vs = vs + 1;

stiva[vs].x = i;

stiva[vs].y = j - 1;

f = true;

}

i = stiva[vs].x;

j = stiva[vs].y;

k = k - 1;

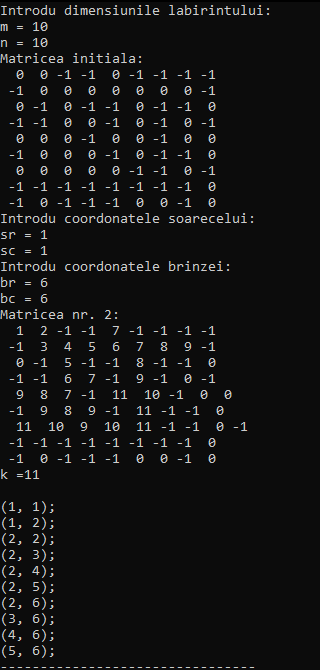
} while (k != 1);

for (i = vs; i > 1; i--)

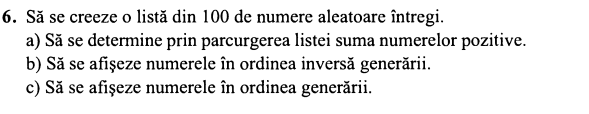
cout << "\n(" << stiva[i].x << ", " << stiva[i].y << ");";

return 0;

}



**Lucrare de laborator Nr.6**



#include <iostream>

#include<time.h>

#include<stdlib.h>

#define MAX\_NUM 100

using namespace std;

struct num

{

int nr;

};

int main() {

int n = 100, random, sum = 0, sumn = 0;

num list[n];

srand(time(NULL));

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

{

random = rand() % MAX\_NUM;

list[i].nr = random;

}

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

{

if (list[i].nr >= 0) sum += list[i].nr;

else if (list[i].nr <= 0) sumn += list[i].nr;

}

cout << "a) Suma numerelor pozitive din lista : " << sum << endl;

cout << "a\*) Suma numerelor negative din lista : " << sumn << endl;

cout << endl << "b) Lista generata in ordine inversa : " << endl;

for (int i = n; i >= 1; i--)

{

cout << list[i].nr << " ";

}

cout << endl;

cout << endl << "c) Lista in ordinea generarii : " << endl;

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

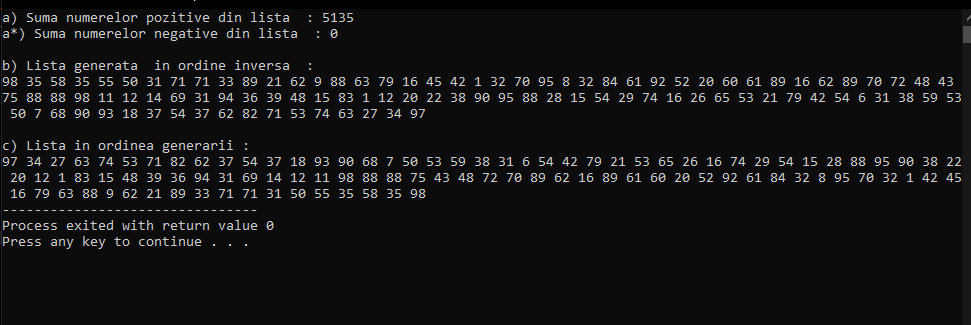
{

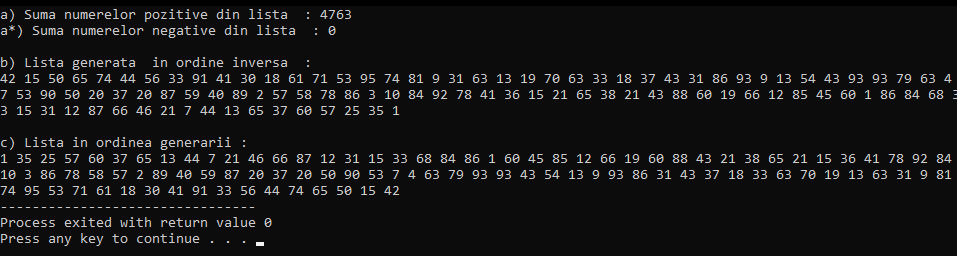
cout << list[i].nr << " ";

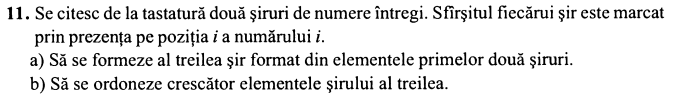
}

return 0;

}







#include<iostream>

using namespace std;

struct read\_array\_response {

int x[100];

int length;

};

typedef struct read\_array\_response array;

array read\_array(string s) {

array res;

int i = 0;

cout << "Introduceti "<< s << " sir: ";

do {

i++;

cin >> res.x[i];

} while(res.x[i] != i);

res.length = i;

return res;

}

array append(array x, array y, int start = 0) {

x.length += y.length;

for (int i = 1; i <= y.length; i++) {

x.x[i + start] = y.x[i];

}

return x;

}

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void bubbleSort(int \*array, int size) {

for(int i = 1; i<=size; i++) {

int swaps = 0;

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

if(array[j] > array[j+1]) {

swap(array[j], array[j+1]);

swaps = 1;

}

}

if(!swaps)

break;

}

}

void print\_array(array x) {

for (int i = 1; i <= x.length; i++) {

cout << x.x[i] << " ";

}

}

int main() {

array a, b, c;

a = read\_array("primul");

b = read\_array("al doilea");

cout << "Sirul concatinat este: " << endl;

c = append(c, a, 0);

c = append(c, b, a.length);

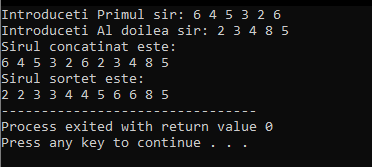
print\_array(c);

cout << endl << "Sirul sortat este: " << endl;

bubbleSort(c.x, c.length);

print\_array(c);

}



**Lucrare de laborator Nr.7**

**Problemă. Se dă V vectorul-heap și nodul I. De scris un algoritm care utilizând reprezentarea Heap:**

**a)De aflat elemental maxim din M;**

**b) De ordonat crescător vectorul M.**

#include <iostream>

#include <fstream>

using namespace std;

ifstream fin("vector.in");

int max(int a[], int n){

int max = a[0];

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

if (a[i] > max)

max = a[i];

return max;

}

int min(int a[], int n){

int min = a[0];

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

if (a[i] < min)

min = a[i];

return min;

}

void creare\_heap(int a[],int n){

int i, j;

int aux;

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

for (j = 0;j < n/2; j++){

if (a[j] < a[2\*j+1]){

aux = a[j];

a[j] = a[2\*j+1];

a[2\*j+1] = aux;

}

if (a[j] < a[2\*j+2]){

aux = a[j];

a[j] = a[2\*j+2];

a[2\*j+2] = aux;

}

}

}

void sortare\_desc(int a[], int heap[],int &n, int &k){

int i;

int f = 0;

cout << "Max:" << a[0] << endl;

while (n){

int val = max(a,n);

int aux;

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

if (a[i] == val && i != n-1){

aux = a[0];

a[0] = a[i];

a[i] = aux;

}

aux = a[0];

a[0] = a[n-1];

a[n-1] = aux;

if (a[n-1] == val){

heap[k] = a[n-1];

n--; k++;

}

if (n == 0)

cout << "Min:" << a[n] << endl;

}

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

cout << heap[i] << ' ';

}

void sortare\_cresc(int a[], int heap[],int &n, int &k){

int i;

int f = 0;

cout << "Max:" << a[0] << endl;

while (n){

int val = min(a,n);

int aux;

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

if (a[i] == val && i != n-1){

aux = a[0];

a[0] = a[i];

a[i] = aux;

}

aux = a[0];

a[0] = a[n-1];

a[n-1] = aux;

if (f == 0 && a[n-1] == val){

cout << "Min:" << a[n-1] << endl;

}

if (a[n-1] == val){

heap[k] = a[n-1];

n--; k++;

f++;

}

}

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

cout << heap[i] << ' ';

cout << endl;

}

void problema2(int arr[], int m){

int ind;

cout << "Introduceti pozitia nodului carui vreti sa aflati informatie :";

cin >> ind;

if (ind >= 1){

cout << "Parintele nodului " << arr[ind] << " este nodul: " << arr[ind/2] << endl;

}

else

cout << "Nodul dat este radacina arborelui.\n";

if (ind\*2+1 < m){

cout << "Descendentul stang " << arr[ind];

cout << " este nodul: " << arr[ind\*2+1] << endl;

}

else

cout << "Nu are descendent stang\n";

if (ind\*2+2 < m){

cout << "Descendentul drept " << arr[ind];

cout << " este nodul: " << arr[ind\*2+2] << endl;

}

else

cout << "Nu are descendent drept.\n";

cout << "\n";

}

int main(){

int a[100];

int heap[100];

int k = 0;

int i, n;

int optiuni;

fin >> n;

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

fin >> a[i];

cout << "Vector citit:\n";

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

cout << a[i] << ' ';

cout << endl;

cout << "Problema 1" << endl;

cout << "Vector heap:\n";

creare\_heap(a,n);

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

cout << a[i] << ' ';

cout << endl;

cout << "Problema 2" << endl;

problema2(a,n);

cout << "Problema 3" << endl;

cout << "Alegeti cifra care va trebuie\n";

cout << "1.Sortare descrescatoare\n";

cout << "2.Sortare crescatore\n";

cin >> optiuni;

switch(optiuni){

case 1 : {

sortare\_desc(a,heap,n,k);

}; break;

case 2 : {

sortare\_cresc(a,heap,n,k);

}; break;

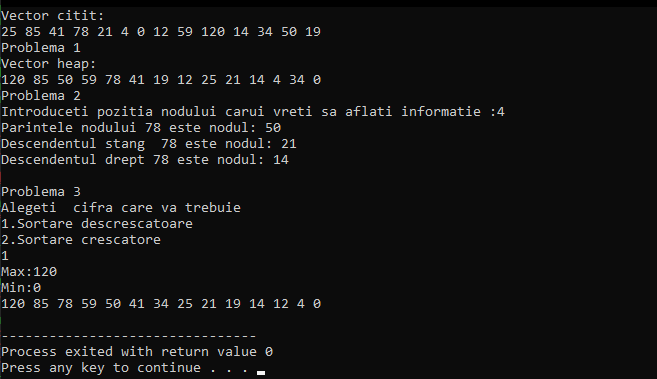
default :

cout << "A-ti ales cifra gresita";

}

cout << endl;

return 0; }



vector.in contine :



**Lucrare de laborator Nr.8**



#include <iostream>

using namespace std;

int sum(int n){

if (n == 0)

return 0;

else

return n + sum(n-1);

}

int main()

{

int n;

cout<<"Dati n = : ";

cin>>n;

cout << "Suma numerelor pina la n este : "<<sum(n) << endl;

return 0;

}

