**P1.** Scrieti testele (assert-urile) care verifica daca functia rmatoare ruleaza corect. (minim 3 teste relevante)

float maxim(float a, float b, float c);

|  |  |
| --- | --- |
| **Test** | **Cazul pe care “il prinde”** |
| assert(maxim(1,2,3)==3) | Uzual |
| assert(maxim(2,2,3)==3) | 2 valori egale, max al treilea |
| assert(maxim(2,2,1)==2) | 2 valori egale, max e unul din ele |
| assert(maxim(2,2,2)==2) | 3 valori egale |
| assert(maxim(-1,-2,-3)==-1) | Valori negative |

**P2.** Ce afiseaza urmatorul program?

|  |  |
| --- | --- |
| **Cod** | **Afiseaza** |
| #include <iostream>  using namespace std;  int main() {  int pas = 1;  for (int i = 1; i <= 20; i+=pas) {  cout << i << " ";  pas++;  }  } | |  |  | | --- | --- | | i | pas | | 1 | 1 | | 3 | 2 | | 6 | 3 | | 10 | 4 | | 15 | 5 | | 21 | 6 | |  |  | |  |  |   1 3 6 10 15 |

**P3.** Ce afiseaza urmatorul program? Scrieti apelurile recursive si rezultatul final. Nu uitati de variabila globala!

|  |  |
| --- | --- |
| **Cod** | **Rulare** |
| #include <iostream>  using namespace std;  int a = 0;  int f(int x, int y) {  a++;  cout << x << " " << y << endl;  if (x < y) {  return x + y + f(2 \* x, y - 1);  }  return 0;  }  int main() {  cout << f(1, 10) << " ";  cout << a;  } | a = ~~0 1 2 3~~ 4  f(1,10) = 1 + 10 + f(2,9) = 34  f(2,9) = 2 + 9 + f(4,8) = 23  f(4,8) = 4 + 8 + f(8,7) = 12  f(8,7) = 0  1 10 2 9 4 8 8 7 34 4 |

**P4.** Implementati functia care, primind ca parametru un vector si lungimea acestuia, returneaza intervalul minim in care sunt cuprinse toate valorile vectorului. De exemplu, daca vectorul contine valorile {3,4,1,7,9,2} se va returna intervalul [1,9].

Interval f(int a[], int n);

Definiti structura necesara, iar in functia main dati un exemplu sugestiv de utilizare a functiei.

#include <iostream>

#include <fstream>

#include <cassert>

using namespace std;

struct Interval {

int min, max;

};

Interval f(int a[], int n);

int main() {

int a[10] = { 3,4,1,7,9,2 };

int n = 6;

Interval interval = f(a, n);

cout << interval.min << " " << interval.max;

}

Interval f(int a[], int n) {

Interval t;

t.min = INT\_MAX;

t.max = INT\_MIN;

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

{

if (a[i] < t.min) {

t.min = a[i];

}

if (a[i] > t.max) {

t.max = a[i];

}

}

return t;

}

**P5.** Implementati functia „Genereaza” si „Afiseaza”, in asa fel incat urmatoarea functie main sa fie valida.

int main(){

int a[50][50]; int n;

Genereaza(a, n);

Afiseaza(a,n);

}

„Genereaza” este o functie care, primind ca parametru o matrice patratica si dimensiunea acesteia, citeste de la tastatura dimensiunea matricii, apoi o completeaza ca in exemplul urmator.

n=5

1 0 1 0 1

2 0 2 0 2

3 0 3 0 3

4 0 4 0 4

5 0 5 0 5

„Afiseaza” va scrie pe ecran matricea, intr-o forma similara exemplului.

00 01 02 03 04

10 11 12 13 14

20 21 22 23 24

30 31 32 33 34

40 41 42 43 44

void Afiseaza(int a[][50], int n) {

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

{

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

{

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

}

cout << endl;

}

}

void Genereaza(int a[][50], int n) {

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

{

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

{

if (j % 2 == 0) {

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

}

else {

a[i][j] = 0;

}

}

}

}

**P6.** Implementati, utilizand pointeri (fara indecsi), functia care, primind ca parametru un sir de caractere continand litere mici ale alfabetului englez, inlocuieste vocalele cu spatii.

int main() {

char S[100] = "treimasini";

cout << S << endl;

VocaleSpatii(S);

cout << S;

}

**O imagine care conține text, Font, captură de ecran, număr

Descriere generată automat**

#include <iostream>

#include <fstream>

#include <cassert>

using namespace std;

void VocaleSpatii2(char S[]) {

for (int i = 0; S[i]; i++)

{

if (strchr("aeiou", S[i])) {

S[i] = ' ';

}

}

}

void VocaleSpatii(char S[]) {

for (char\* p = S; \*p; p++)

{

if (strchr("aeiou", \*p)) {

\*p = ' ';

}

}

}

int main() {

char S[100] = "treimasini";

cout << S << endl;

VocaleSpatii(S);

cout << S;

}

**P7.** Fie structura:

struct Vector {

int a[100];

int n;

};

Implementati functia void GenereazaPatrate(Vector &v); care citeste de la tastatura lungimea vectoului, apoi il completeaza cu patrate perfecte, ca in exemplu {1,4,9,16, ...}.

#include <iostream>

#include <fstream>

#include <cassert>

using namespace std;

struct Vector {

int a[100];

int n;

};

void GenereazaPatrate2(Vector& v);

void GenereazaPatrate(Vector& v);

int main() {

Vector v;

GenereazaPatrate(v);

for (int i = 0; i < v.n; i++)

{

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

}

}

void GenereazaPatrate2(Vector& v) {

v.n = 10;

for (int i = 0; i < v.n; i++)

{

v.a[i] = (i + 1) \* (i + 1);

}

}

void GenereazaPatrate(Vector& v) {

v.n = 0;

for (int x = 1; x <= 10; x++)

{

v.a[v.n++] = x \* x;

}

}

**P8.** Se citeste n de la tastatura. Utilizand un vector alocat dinamic, cititi n valori de la tastatura. Determinati minimul valorilor din prima jumatate a vectorului, maximul valorilor din a doua jumatate, apoi dealocati vectorul.

#include <iostream>

using namespace std;

int main() {

int\* a;

int n;

cin >> n;

a = new int[n];

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

{

cin >> a[i];

}

int min = INT\_MAX;

for (int i = 0; i < n / 2; i++)

{

if (a[i] < min) {

min = a[i];

}

}

int max = INT\_MIN;

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

{

if (a[i] > max) {

max = a[i];

}

}

cout << min << " " << max;

delete[]a;

}

**P9.** Implementati o functie care, primind ca parametru un vector, lungimea acestuia si un pointer la o functie, modifica valorile din vector prin „aplicarea” functiei venite ca parametru pe fiecare valoare din vector. Implementati inclusiv functiile Jumatate si SchimbaSemn.

Urmatorul „main” ar trebui sa fie valid:

int main() {

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat int a[100] = { 10,-20,30,40,- 50 };

int n = 5;

Afis(a, n);

f(a, n, Jumatate);

Afis(a, n);

f(a, n, SchimbaSemn);

Afis(a, n);

}

#include <iostream>

using namespace std;

void Afis(int a[], int n) {

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

{

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

}

}

int Jumatate(int n) {

return n / 2;

}

int SchimbaSemn(int n) {

return -n;

}

void f(int a[], int n, int Schimbare(int)) {

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

{

a[i] = Schimbare(a[i]);

}

}

int main() {

int a[100] = { 10,-20,30,40,-50 };

int n = 5;

Afis(a, n);

f(a, n, Jumatate);

Afis(a, n);

f(a, n, SchimbaSemn);

Afis(a, n);

}