Tema Lab. 4

NICOLETA RADU

# Exericitiul 1

float aria(int r, float pi)

{

float a;

a = pi \* r \* r;

return a;

}

int main()

{

int raza;

float pi;

float A;

cout << "Dati raza: ";

cin >> raza;

pi = 3.14f;

//modulul apelator

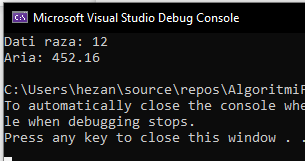
A = aria(raza, pi);

cout << "Aria: " << A << '\n';

system("pause");

return 0;

}



b.

float aria(int r, float pi)

{

float a;

a = pi \* r \* r;

return a;

}

int main()

{

int raza;

float pi;

cout << "Dati raza: ";

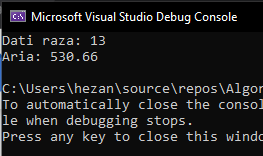
cin >> raza;

pi = 3.14f;

cout << "Aria: " << aria(raza,pi) << '\n';

system("pause");

return 0;



c.

float aria(int r, float pi=3.14f)

{

float a;

a = pi \* r \* r;

return a;

}

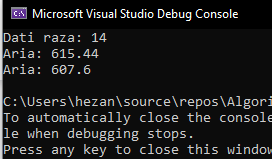
int raza;

cout << "Dati raza: ";

cin >> raza;

cout << "Aria: " << aria(raza) << '\n';

cout << "Aria: " << aria(raza, 3.1f) << '\n';



d.

float aria(int, float = 3.14f);

int main()

{

int raza;

cout << "Dati raza: ";

cin >> raza;

//apelul functiei

cout << "Aria: " << aria(raza) << '\n';

system("pause");

return 0;

}

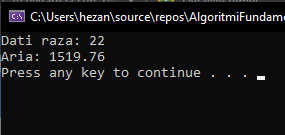
//corpul functiei

float aria(int r, float pi)

{

return pi \* pow(r,2);

}



# Exercitiul 2

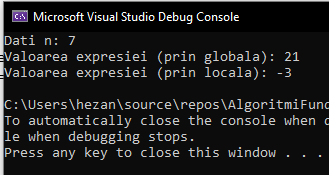
Variabile globale si locale.

Se vor scrie 2 funcții:

* Prima pentru calculul sumei: 1+2+3+…+n
* A doua pentru calculul sumei: 1-2+3-4+…±n.

Pentru prima funcție rezultatul se va reține într-o variabilă declarată global, iar pentru a doua funcție rezultatul va fi transmis prin rezultatul acesteia.

La apel, atat variabila rezultat pentru prima, cât și variabila rezultat pentru a doua vor avea același nume.



int suma1(int n)

{

for (size\_t i = 1; i < n; i++)

{

suma += i;

}

return suma;

}

int suma2(int n)

{

int suma{ 0 };

for (size\_t i = 1; i < n; i++)

{

if (i % 2 == 1)

{

suma = suma + i;

}

if (i % 2 == 0)

{

suma = suma - i;

}

}

return suma;

}

int main()

{

int n;

do {

cout << "Dati n: ";

cin >> n;

} while (n <= 0);

suma1(n);

int suma;

suma = suma2(n);

//afisez cele doua rezultate

cout << "Valoarea expresiei (prin globala): " << ::suma << '\n';

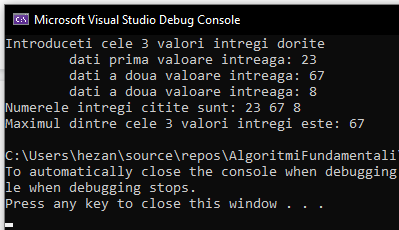
cout << "Valoarea expresiei (prin locala): " << suma << '\n';

return 0;

}

# Exercitiul 3

Se vor construi functii pentru citirea a trei numere intregi si se va determina maximul dintre cele 3 valori numerice citite.



//functie de citire a unui numar intreg cu transmitere rezultat prin tipul rezultat

int citire1()

{

int i;

cout << "\tdati prima valoare intreaga: ";

cin >> i;

return i;

}

//functie ptr. citirea unui numar intreg, cu rezultat transmis prin linia de parametri (transfer prin adrese)

void citire2(int\* i)

{

cout << "\tdati a doua valoare intreaga: ";

cin >> \*i;

}

//functie ptr. citirea unui numar intreg, cu rezultat transmis prin linia de parametri (transfer prin referinte)

void citire3(int& i)

{

cout << "\tdati a doua valoare intreaga: ";

cin >> i;

}

//functie de afisare a unei variabile intregi

void afisare(int i)

{

cout << i << ' ';

}

//functie pentru determinarea valorii maxime intre doua intregi

int maxim(int i, int j)

{

int m;

i >= j ? m = i : m = j;

return m;

}

/\* o alta varianta a functiei de maxim:

int maxim(int i, int j)

{

if(i>=j)

return i;

return j;

}

\*/

//functia principala in rulare

int main()

{

int n1, n2, n3;

//citirea celor 3 numere intregi

cout << "Introduceti cele 3 valori intregi dorite\n";

n1 = citire1(); //apelul functiei 'citire1'

citire2(&n2); //apelul functiei 'citire2'

citire3(n3); //apelul functiei 'citire3'

//afisarea celor 3 numere intregi citite

cout << "Numerele intregi citite sunt: ";

afisare(n1);

afisare(n2);

afisare(n3);

cout << '\n';

//determinarea valorii maxime intre cele 3 numere intregi citite

int m = maxim(n1, n2);

if (n3 > m)

m = n3;

cout << "Maximul dintre cele 3 valori intregi este: " << m << '\n';

system("pause");

return (0);

}

# Exercitiul 4

**Exercitiu 4.** Se va scrie o functie, care primind un parametru n, va afisa secventa de mai

jos:

1

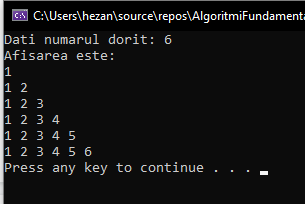
1 2

1 2 3

......

1 2 3 ... n

Se va scrie programul C++ complet, cu exemplu de apel al functiei respective.



void afisare(int n)

{

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

{

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

cout << j << ' ';

cout << '\n';

}

}

int main()

{

int n;

do

{

cout << "Dati numarul dorit: ";

cin >> n;

} while (n < 1);

cout << "Afisarea este:\n";

afisare(n);

system("pause");

return (0);

}

# Exercitiul 5

**Exercitiu 5.** Fără a rula rutina de mai jos, sa se determine rezultatul afisat

1. cout << a << ' ' << b << '\n';   //6 5

cout << f(a, b) << '\n';     //21

cout << a << ' ' << b << '\n';   //6 5

1. se va modifica antetul de functie, astfel incat transferul de parametri sa se faca prin adrese.

int f(int\* m, int\* n)   //\*m=6 \*n=5

{

\*m = \*n + b;   //\*m=5+5=10 (a)

\*n += 1;       //\*n=5+1=6 (b)

return \*n + b + \*m;  //6+6+10=22

}

void main()

{

a = 6;

b = 5;

cout << a << ' ' << b << '\n';   //6 5

cout << f(&a, &b) << '\n';     //22

cout << a << ' ' << b << '\n';   //10 6

system("pause");

1. se va modifica antetul de functie, astfel incat transferul de parametri sa se faca prin referinte

#include<iostream>

using namespace std;

int a, b;

int f(int& m, int& n)   //m=6 n=5

{

m = n + b;   //m=5+5=10 (a)

n += 1;       //n=5+1=6 (b)

return n + b + m;  //6+6+10=22

}

void main()

{

a = 6;

b = 5;

cout << a << ' ' << b << '\n';   //6 5

cout << f(a, b) << '\n'; //22

cout << a << ' ' << b << '\n';   //10 6

system("pause");

}

# Exercitiul 6

// **Exercitiu 6.** Scrierea functiilor cu prototip

/\*

Se vor construi functii pentru:

- citirea unui numar real

- afisarea unui numar real

- interschimbarea a doua valori reale:

- varianta cu transfer prin valori;

- varianta cu transfer prin adrese;

- varianta cu transfer prin referinte.

\*/

#include<iostream>

using namespace std;

//citirea unui numar real

float citire(const char\*); //varianta 1

void citire(float\*, const char\*);  //varianta 2

//afisarea unui numar real

void afisare(float, const char\*);

//interschimbarea a doua valori - functie cu transfer parametri prin valoare

void schimba1(float, float);

//interschimbarea a doua valori - functie cu transfer parametri prin adrese

void schimba2(float\*, float\*);

//interschimbarea a doua valori - functie cu transfer parametri prin referinte

void schimba3(float&, float&);

//varianta 1 de citire (cu transmitere prin tipul rezultat)

float citire(const char\* s)

{

float n;

cout << "Dati " << s << ": ";

cin >> n;

return n;

}

//varianta 2 de citire (cu transmitere prin linia de parametri)

void citire(float\* n, const char\* s)

{

cout << "Dati " << s << ": ";

cin >> \*n;

}

//afisarea numerelor reale

void afisare(float n, const char\* s)

{

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

}

//interschimbarea valorilor prin cele 3 functii

void schimba1(float i, float j)

{

float t;

t = i;

i = j;

j = t;

}

void schimba2(float\* i, float\* j)

{

\*i = \*i + \*j;

\*j = \*i - \*j;

\*i = \*i - \*j;

}

void schimba3(float& i, float& j)

{

swap(i, j);

}

//functia principala in rulare

int main()

{

float a, b;

a = citire("primul numar");  //functia de citire cu transmitere prin rezultat

citire(&b, "al doilea numar"); //functia de citire cu transmitere prin parametri

//afisarea numerelor reale citite

cout << "Valorile initiale:\n";

afisare(a, "\tprimul numar");

afisare(b, "\tal doilea numar");

// dupa interschimbarea prin transfer valori

schimba1(a, b);

cout << "Dupa interchimbarea prin valori:\n";

afisare(a, "\tprimul numar");

afisare(b, "\tal doilea numar");

//dupa interschimbarea prin transfer adrese

schimba2(&a, &b);

cout << "Dupa interchimbarea prin adrese:\n";

afisare(a, "\tprimul numar");

afisare(b, "\tal doilea numar");

//dupa interschimbarea prin transfer referinte

schimba3(a, b);

cout << "Dupa interchimbarea prin referinte:\n";

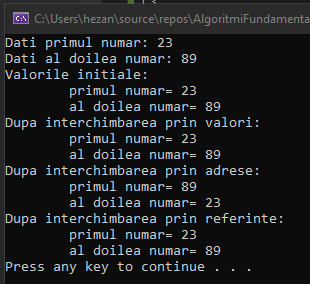
afisare(a, "\tprimul numar");

afisare(b, "\tal doilea numar");

system("pause");

return (0);

}



# Exercitiul 7

**Exercitiu 7.** Definiti o functie, care primeste ca parametru un numar intreg, si returneaza

cel mai apropiat numar prim, mai mare ca acesta. Daca numarul este prim, il afiseaza pe

acela.

Exemplu: cout<<f(20); va afisa valoarea 23

bool f(int n)

{

bool x = true;

if (n == 0 or n == 1 or (n%2==0 and n!=2))

{

x = false;

return x;

}

if (n == 2)

{

x = true;

return x;

}

for (int i = 3; i \* i <= n; i+=2)

if (n % i == 0)

{

x = false;

return x;

}

return x;

}

int main()

{

int n, i;

cout << "Dati numarul dorit: ";

cin >> n;

i = n;

i++;

while (f(i) == false)

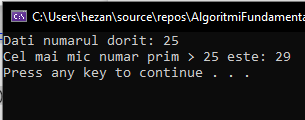
i++;

cout << "Cel mai mic numar prim > " << n << " este: " << i << '\n';

system("pause");

return 0;

}



# Exercitiul 8

**Exercitiu 8.** Scrieti un subprogram care afiseaza numere de 4 cifre care au ultimele 3

cifre egale.

\*/

#include<iostream>

using namespace std;

void f(int n)

{

if (n % 10 == n / 10 % 10)

if (n / 10 % 10 == n / 100 % 10)

cout << n << ' ';

}

int main()

{

int n;

cout << "Numerele cu 3 cifre egale sunt:\n";

for (int n = 1000; n <= 9999; n++)

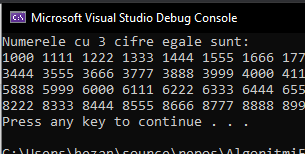
f(n);

cout << '\n';

system("pause");

return 0;

}



# Exercitiul 9

**Exercitiu 9.** Scrieti un subprogram care să afişeze numerele pare mai mici ca n care au si

prima cifra para.

\*/

#include<iostream>

using namespace std;

int nr\_cifre(int n)

{

int x = 0;

while (n != 0)

{

n = n / 10;

x++;

}

return x;

}

int prima\_cifra(int n)

{

return n / pow((double)10, nr\_cifre(n) - 1);

}

int main()

{

int n;

cout << "Dati n: ";

cin >> n;

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

if (i % 2 == 0 && prima\_cifra(i) % 2 == 0)

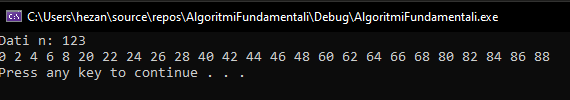
cout << i << " ";

cout << '\n';

system("pause");

return 0;

}



# Exercitiul 10

**Exercitiu 10.** Scrieti o functie care primeste ca parametru un numar intreg si returneaza

suma dintre prima si ultima cifra a numarului.\*/

#include<iostream>

using namespace std;

int nr\_cifre(int n)

{

int x = 0;

while (n != 0)

{

n = n / 10;

x++;

}

return x;

}

int prima\_cifra(int n)

{

return n / pow((double)10, nr\_cifre(n) - 1);

}

int suma(int n)

{

int s;

s = n % 10 + prima\_cifra(n);

return s;

}

int main()

{

int n;

cout << "Dati n: ";

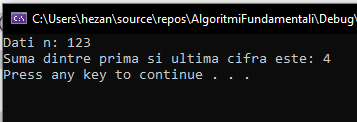
cin >> n;

cout << "Suma dintre prima si ultima cifra este: " << suma(n) << '\n';

system("pause");

return 0;

}



# Exercitiul 11

**Exercitiu 11.** Definiti o functie care primeste ca parametru un numar intreg si care

returneaza media aritmetica a divizorilor acestuia.

\*/

#include<iostream>

using namespace std;

float f(int n)

{

int s = 0, c = 0;

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

if (n % i == 0)

{

s += i;

c++;

}

s += n;

c += 1;

return s / c;

}

int main()

{

int n;

cout << "Dati n: ";

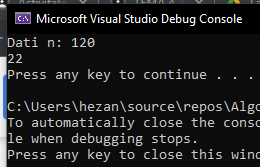
cin >> n;

cout << f(n) << '\n';

system("pause");

return 0;

}



# Exercitiul 12

**Exercitiu 12.** Scrieţi un subprogram, care verifică dacă un număr intreg pozitiv, cu cel

mult 4 cifre, transmis ca parametru este prim sau nu. Subprogramul va returna valoarea

1/true dacă numărul transmis ca parametru este prim şi 0/false în caz contrar.

\*/

#include<iostream>

using namespace std;

//numarul de cifre al unui numar transmis ca parametru

int nr\_cifre(int n) //n este numarul folosit spre prelucrare

{

int x = 0; //nr. de cifre

while (n) //n!=0

{

n /= 10; //n=n/10;

x++;

}

return x;

}

//citirea unui numar intreg (>0 si din maxim 4 cifre)

void citire(int\* x) //x este numarul ce va fi citit

{

do

{

cout << "Introduceti numarul dorit: ";

cin >> \*x;

} while ((\*x <= 0) || (nr\_cifre(\*x) > 4));

}

//verificarea daca un numar este prim sau nu

bool dc\_prim(int n) //n este valoarea testata

{

bool x = true;

if (n == 2)

return x;

if (n == 1 || n % 2 == 0)

{

x = false;

return x;

}

for (int i = 3; i \* i <= n; i += 2)

if (n % i == 0)

{

x = false;

return x;

}

return x;

}

//programul principal

int main()

{

int n;

citire(&n);

if (dc\_prim(n)) //(dc\_prim(n)!=0), adica TRUE

cout << "Numarul " << n << " este prim\n";

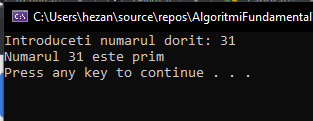
else

cout << "Numarul " << n << " nu este prim\n";

system("pause");

return (0);

}



# Exercitiul 13

**Exercitiu 13.**

Scrieţi un subprogram care returnează suma divizorilor proprii (fără 1 şi el însuşi) ai

unui număr întreg pozitiv, cu cel mult 4 cifre, transmis ca parametru.

\*/

#include<iostream>

using namespace std;

//citirea numarului

int citire()

{

int x; //numarul ce va fi citit

do {

cout << "Introduceti numarul dorit: ";

cin >> x;

} while (x <= 0 || x > 9999);

return x;

}

//suma divizorilor (fara 1 si el insusi)

int suma\_div(int n)

{

int s = 0;

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

if (n % i == 0)

s += i;

return s;

}

//programul principal

int main()

{

int n;

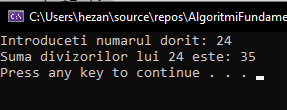
n = citire();

cout << "Suma divizorilor lui " << n << " este: " << suma\_div(n) << '\n';

system("pause");

return (0);

}



# Exercitiul 14

**Exercitiu 14.** Scrieţi un subprogram care returnează cea mai mare cifră a unui număr

natural, cu cel mult 8 cifre, transmis ca parametru.

\*/

#include<iostream>

using namespace std;

//numarul de cifre al unui numar transmis ca parametru

int nr\_cifre(int n) //n este numarul folosit spre prelucrare

{

int x = 0; //nr. de cifre

while (n) //n!=0

{

n /= 10; //n=n/10;

x++;

}

return x;

}

//citirea unui nr. intreg pozitiv de maxim 8 cifre

void citire(long long\* x)

{

do {

cout << "Dati numarul: ";

cin >> \*x;

} while ((\*x <= 0) || (nr\_cifre(\*x) > 8));

}

//cifra maxima a unui numar intreg

int cifra\_maxima(long long n)

{

int x, y;

if (nr\_cifre(n) == 1)

return n;

x = n % 10;

while (n != 0)

{

n = n / 10; //n/=10;

y = n % 10; //n%=10;

if (y > x)

x = y;

}

return x;

}

//functia principala in rulare

void main()

{

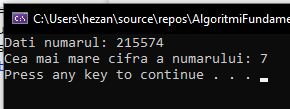
long long n;

citire(&n);

cout << "Cea mai mare cifra a numarului: " << cifra\_maxima(n) << '\n';

system("pause");

}



# Exercitiul 15

**Exercitiu 15.** Scrieţi un subprogram care primeşte printr-un parametru un numar intreg

pozitiv, cu maximum 8 cifre şi returnează suma cifrelor numărului.

\*/

#include<iostream>

using namespace std;

//numarul de cifre al unui numar transmis ca parametru

int nr\_cifre(int n) //n este numarul folosit spre prelucrare

{

int x = 0; //nr. de cifre

while (n) //n!=0

{

n /= 10; //n=n/10;

x++;

}

return x;

}

//citirea unui nr. intreg pozitiv de maxim 8 cifre

void citire(long long\* x)

{

do {

cout << "Dati numarul: ";

cin >> \*x;

} while ((\*x <= 0) || (nr\_cifre(\*x) > 8));

}

//cifra maxima a unui numar intreg

int suma\_cifre(long long n)

{

int s = 0;

if (nr\_cifre(n) == 1)

return n;

while (n != 0)

{

s = s + n % 10;

n /= 10;

}

return s;

}

//functia principala in rulare

void main()

{

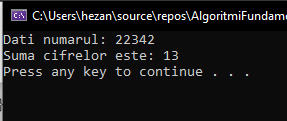
long long n;

citire(&n);

cout << "Suma cifrelor este: " << suma\_cifre(n) << '\n';

system("pause");

}



# Exercitiul 16

**Exercitiu 16.** Scrieţi un subprogram care primeşte printr-un parametru o valoare naturală n, cu maximum 8 cifre şi returnează numărul de cifre pare ale numărului n.

\*/

#include<iostream>

using namespace std;

//numarul de cifre al unui numar transmis ca parametru

int nr\_cifre(int n) //n este numarul folosit spre prelucrare

{

int x = 0; //nr. de cifre

while (n) //n!=0

{

n /= 10; //n=n/10;

x++;

}

return x;

}

//citirea unui nr. intreg pozitiv de maxim 8 cifre

void citire(long long\* x)

{

do {

cout << "Dati numarul: ";

cin >> \*x;

} while ((\*x <= 0) || (nr\_cifre(\*x) > 8));

}

//numarul de cifre pare ale unui numar intreg

int nr\_cifre\_pare(long long n)

{

int y, k = 0;

if (nr\_cifre(n) == 1)

return n;

while (n != 0)

{

y = n % 10;

if (y % 2 == 0)

k++;

n /= 10;

}

return k;

}

//functia principala in rulare

void main()

{

long long n;

citire(&n);

cout << "Numarul de cifre pare dintr-un numar intreg este: ";

cout << nr\_cifre\_pare(n) << '\n';

system("pause");

}

