Tema Lab. 3

NICOLETA RADU

# Aplicatia 1

Valoarea expresiei S=1+2+3+4+…+n.

int n;

long long S=0;

do {

cout << "Dati n: ";

cin >> n;

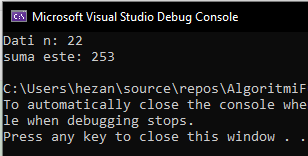
} while (n < 1);

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

S += i;

cout << "suma este: " << S << '\n';

return 0;



# Aplicatia 2

 Se citeste un numar întreg, strict pozitiv. Sa se determine dacă numarul citit este sau nu numar prim.

int n;

do {

cout << "Dati numarul: ";

cin >> n;

} while (n < 1);

if (n == 2)

{

cout << "Numarul este prim\n";

system("pause");

return 1;

}

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

{

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

system("pause");

return 2;

}

bool ok = true;

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

if (n % i == 0)

{

ok = false;

break;

}

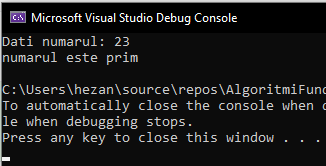
if (ok)

cout << "numarul este prim\n";

else

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

return 0;



# Aplicatia 3

Valoarea expresiei S=1!+2!+3!+4!+…+n!

int n;

do {

cout << "Dati n: ";

cin >> n;

} while (n < 1);

int S = 0, f;

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

{

f = 1;

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

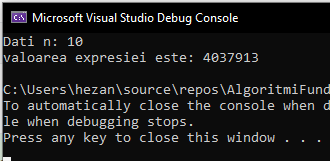
f \*= j;

S += f;

}

cout << "valoarea expresiei este: " << S << '\n';

return 0;



# Aplicatia 4

Se citesc valori intregi, până la întâlnirea valorii 0. Să se realizeze produsul valorilor întâlnite.

int n;

int P = 1;

do {

cout << "Dati n: ";

cin >> n;

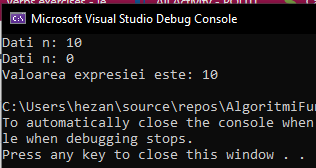
if (n)

P \*= n;

} while (n != 0);

cout << "Valoarea expresiei este: " << P << '\n';

return 0;



# Exercitiul 4

 Să se calculeze următoarea expresie: 1‐2+3‐4+...±n, unde n este un întreg dat de la tastatură, >=1.

int n{0}, rezultat{1}, auxiliar{0};

cout<<"n= ";

cin >> n;

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

{

if(i % 2 == 0)

{

rezultat = rezultat - i;

cout << "i par este: " << i << endl;

} else

{

rezultat = rezultat + i;

cout<<"i impar este : " << i << endl;

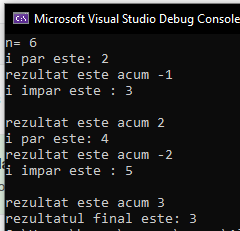
cout<<endl;

}

cout<<"rezultat este acum "<<rezultat<<endl;

}

cout << "rezultatul final este: " << rezultat;



# Exercitiul 5

Se citesc două numere naturale m și n. Calculați nm.

int n{0}, m{0}, produsPutere{1};

cout<<"n = ";

cin>> n;

cout<<"m = ";

cin>>m;

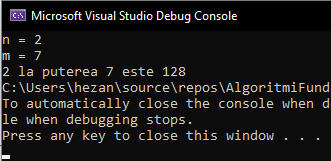
for(int i=0; i<m; i++)

{

produsPutere= produsPutere \* n;

}

cout<<n<<" la puterea "<<m<<" este "<< produsPutere;



# Exercitiul 6

Se citesc două numere întregi. Să se calculeze produsul lor fără a folosi operatorul pentru înmulțire.

int n{0}, m{0}, rezultat{0};

cout<<"n= ";

cin>>n;

cout<<"m= ";

cin>>m;

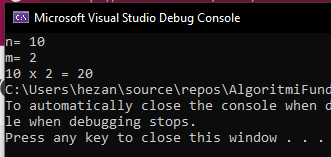
for(int i=0; i<m; i++)

{

rezultat= rezultat + n;

}

cout<< n << " x " << m << " = " << rezultat;



# Exercitiul 7

Se citesc două numere întregi. Sa se calculeze catul si restul impartirii  lui  a la b, fara a folosi operatori pentru cât și rest.

int a, b, i;

cout << "Dati a: ";

cin >> a;

cout << "Dati b: ";

cin >> b;

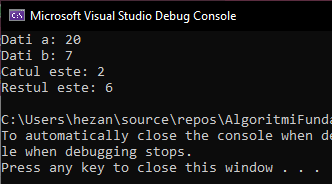
for (i = 0; a >= b; i++)

a = a - b;

cout << "Catul este: " << i << '\n';

cout << "Restul este: " << a << '\n';

return 0;



# Exercitiul 8

Se citesc de la tastatura n numere întregi nenule. Să se calculeze suma celor de rang par si produsul celor de rang impar

int size{0}, sum{0}, product{1};

cout<<"Introduceti marimea vectorului: ";

cin>> size;

vector <int> myVector (size,0);

for(int i=0; i<myVector.size(); i++)

{

cout<<"At ["<<i<<"] = ";

cin>>myVector.at(i);

}

cout<<endl;

cout<<"Printing Vector"<<endl;

for(int i=0; i<myVector.size(); i++)

{

cout<<"At ["<<i<<"] = "<<myVector.at(i)<<endl;

}

for(int i=0; i<myVector.size(); i++)

{

if(i % 2 == 0)

{

sum += myVector.at(i);

} else

{

product \*= myVector.at(i);

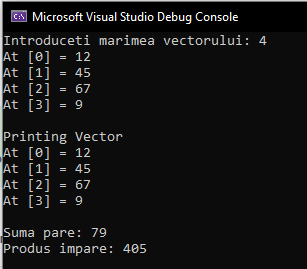
}

}

cout<<endl;

cout<<"Suma pare: "<< sum << endl;

cout<<"Produs impare: "<< product << endl;



# Exercitiul 9

Se citește un numar natural n. Să se determine toate perechile de numere naturale a și b care verifică relația a2+b2=n.

int n{0}, a{0}, b{0};

cout<<"n= ";

cin>>n;

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

{

a = i \* i;

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

{

b = j \* j;

if(a + b == n)

{

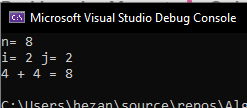
cout<<"i= "<<i<<" j= "<<j<<endl;

cout<<a<<" + "<<b<<" = "<<n<<endl;

}

}

}



# Exercitiul 10

Calculați suma: S=12-22+32-42..(-1)n+1\*n2

int n{0}, rezultat{1}, auxiliar{0};

cout<<"n= ";

cin >> n;

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

{

if(i % 2 == 0)

{

rezultat = rezultat - (i\*i);

cout << "i par este: " << i << endl;

} else

{

rezultat = rezultat + (i\*i);

cout<<"i impar este : " << i;

cout<<endl;

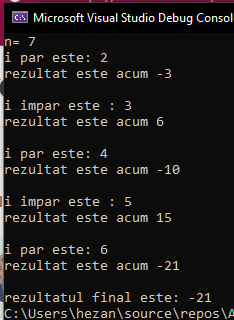
}

cout<<"rezultat este acum "<<rezultat<<endl;

cout<<endl;

}

cout << "rezultatul final este: " << rezultat;



# Exercitiul 11

Se citesc mai multe numere,pana cand utimul numar citit este zero. Să se afle cate numere sunt pozitive si cate sunt negative

int n;

int P = 1;

do {

cout << "Dati n: ";

cin >> n;

} while(n==0);

while (n != 0)

{

P \*= n;

cout << "Dati n: ";

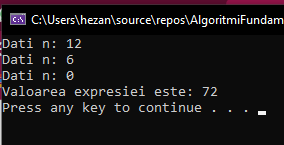
cin >> n;

}

cout << "Valoarea expresiei este: " << P << '\n';

system("pause");

return 0;



# Exercitiul 12

Să se scrie un algoritm care verifică dacă un numar n este prim

int n;

do {

cout << "Dati numarul: ";

cin >> n;

} while (n < 1);

if (n == 2)

{

cout << "Numarul este prim\n";

system("pause");

return 1;

}

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

{

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

system("pause");

return 2;

}

bool ok = true;

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

if (n % i == 0)

{

ok = false;

break;

}

if (ok)

cout << "Numarul este prim\n";

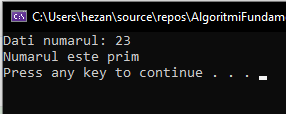
else

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

system("pause");

return 0;

}



# Exercitiul 13

Se dă un numar natural n>5. Să se scrie un algoritm  care returnează toate numerele patrate  perfecte mai mici decat n, dar impare.

int main()

{

int n{ 0 }, produs{ 0 }, j{6};

cout << "n = ";

cin >> n;

n = sqrt(n);

while (sqrt(produs) < n)

{

produs = j \* j;

j++;

if (produs % 2 == 1)

{

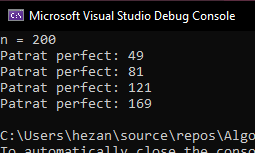
cout << "Patrat perfect: " << produs << endl;

}

}

return 0;

}



# Exercitiul 14

Se dă un numar n. Sa se scrie un algoritm care calculeaza si returneaza suma cifrelor impare din numar.

int n{0}, suma{0};

cout<<"n= ";

cin>>n;

if(n==0)

{

cout<<"Nu se poate";

}

while(n!=0)

{

if((n%10)%2==1)

{

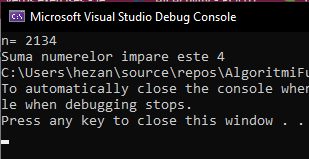
suma += n%10;

}

n = n / 10;

}

cout<<"Suma numerelor impare este "<<suma;



# Exercitiul 15

int n{0}, ultimaCifra{0}, count{0};

cout<<"n= ";

cin>>n;

ultimaCifra= n % 10;

if(n==0)

{

cout<<"Nu se poate";

}

while(n!=0)

{

if(n%10 == ultimaCifra)

{

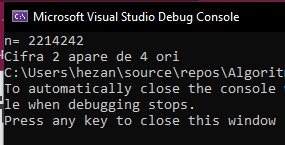
count++;

}

n = n/10;

}

cout<<"Cifra "<<ultimaCifra<<" apare de "<<count<<" ori";



# Exercitiul 16

Se va determina valoarea expresiei S: 1+(1+2)+(1+2+3)+…(1+2+…+n), unde n este un numar întreg, >=1, dat de la tastatură

# Exercitiul 17

Se citește numărul natural n. Să se afișeze următoarea piramidă de numere:

1

1 2

1 2 3

.......

1 2 3 ... n

int i, j, rows;

cout << "Numarul de randuri: ";

cin >> rows;

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

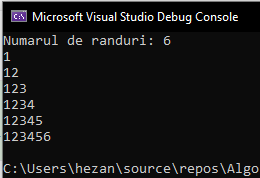
for (j = 1; j <= i; ++j) {

cout << j;;

}

cout << endl;

}



# Exercitiul 18

Se citește numărul natural n, şi un caracter c. Să se afișeze următoarea piramidă:

int i, j, n;

char var = 'c';

cout << "Numarul de randuri: ";

cin >> n;

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

for (int j = 0; j <= i; j++) {

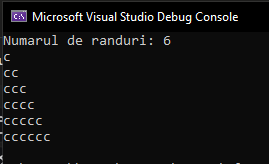
cout << var;

}

cout << endl;

}

return 0;



# Exercitiul 19

Se citește numărul natural n, şi un caracter c. Să se afișeze următoarea piramidă:

c

cc

ccc

....

ccc...c

....

ccc

cc

int i, j, n;

char var = 'c';

cout << "Numarul de randuri: ";

cin >> n;

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

for (int j = 0; j <= i; j++) {

cout << var;

}

cout << endl;

}

i = n, j = 0;

cout << endl;

while (i > 0) {

while (j < i) {

cout << var;

j++;

}

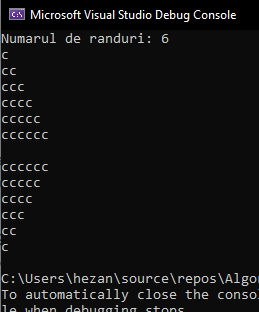
j = 0;

i--;

cout << endl;

}

return 0;



# Exercitiul 20

Se citește numărul natural n, şi două caractere c şi d. Să se afișeze următorul pătrat, format din n linii şi n coloane:

ccc...cc

cdd...dc

....

cdd...dc

ccc...cc

# Exercitiul 21

Se citește numărul natural impar n, şi două caractere c şi d. Să se afișeze următorul pătrat, format din n linii şi n coloane, în care cele două caractere alternează:

cdc...dc

dcd...cd

....

dcd...cd

cdc...dc

# Exercitiul 22

Suma cifrelor unui număr.

int suma{ 0 }, n{ 0 };

cout << "n = ";

cin >> n;

if (n == 0)

{

cout << suma;

}

while (n!=0)

{

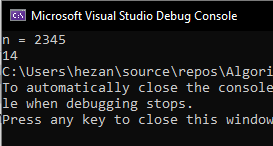
suma += n % 10;

n = n / 10;

}

cout << suma;

return 0;



# Exercitiul 23

Inversul unui număr.

int invers{ 0 }, n{ 0 }, p{ 1 };

cout << "n = ";

cin >> n;

while (n!=0)

{

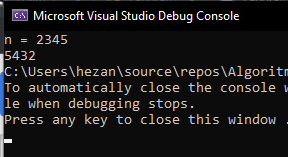
invers = invers \* 10 + (n % 10);

n = n / 10;

}

cout << invers;

return 0;



# Exercitiul 24

Se citește un text ce contine diverse mesaje. Citirea se va face pana la întâlnirea caracterelui ‚.’, cerându‐se contorizarea apariției fiecarei vocale și a tuturor consoanelor întâlnite

char k;

int a, e, i, o, u, c;

a = e = i = o = u = c = 0;

cout << "Introduceti un mesaj: ";

do {

cin >> k;

switch (k)

{

case 'a':

case 'A': a++; break;

case 'e':

case 'E': e++; break;

case 'i':

case 'I': i++; break;

case 'o':

case 'O': o++; break;

case 'u':

case 'U': u++; break;

default:

if ((k >= 'b' and k <= 'z') or (k >= 'B' and k <= 'Z'))

c++;

};

} while (k != '.');

cout << "Numarul de aparitii a|A: " << a << '\n';

cout << "Numarul de aparitii e|E: " << e << '\n';

cout << "Numarul de aparitii i|I: " << i << '\n';

cout << "Numarul de aparitii o|O: " << o << '\n';

cout << "Numarul de aparitii u|U: " << u << '\n';

cout << "Numarul de aparitii consoane: " << c << '\n';

system("pause");

return 0;

# Exercitiul 25

Să se transforme un număr din baza 10 într-o bază **b** oarecare, cu proprietatea **2<=p<=10**.

int baza{ 0 }, numar{ 0 }, rezultat{ 0 }, rest{ 0 }, copyNum{ 0 }, p{1};

cout << "Numarul este: ";

cin >> numar;

copyNum = numar;

cout << "Baza este: ";

cin >> baza;

cout << "Transformarea numaralui " << numar << " din baza 10 in baza " << baza << endl;

cout << endl;

while (numar > baza)

{

rest = numar % baza;

cout << "restul este acum: " << rest << endl;

cout << endl;

numar = numar / baza;

cout << "numar este acum: " << numar << endl;

cout << endl;

rezultat = rest \* p + rezultat;

p = p \* 10;

cout << "rezultat este acum: " << rezultat << endl;

cout << endl;

}

rezultat = numar \* p + rezultat;

cout << endl;

cout << "Transformarea numaralui " << copyNum << " din baza 10 in baza " << baza << " este: " << rezultat << endl;

return 0;

