Tema Laborator 2

NICOLETA RADU

# Exercitiul 1

Ecuatia de gradul 2.

int main()

{

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

float x1{0.0}, x2{0.0}, delta{0.0};

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

if (a == 0)

{

if (b == 0)

{

if (c == 0)

{

cout << "Ecuatie de gradul 0. Solutii infinite";

}

else

{

cout << "Ecuatie de gradul 0. Solutii multimea vida";

}

}

else

{

x1 = (float)-c / b;

cout << "Ecuatie de gradul 1. Solutia este:" << x1;

}

}

else

{

delta = b \* b - 4 \* a \* c;

if (delta < 0)

{

cout << "Ecuatie de gradul 2. Solutii din C";

}

else

{

if (delta == 0)

{

x1 = -b / (2 \* a);

cout << "Ecuatie de gradul 0. Solutie unica: " << x1;

}

else

{

x1 = (-b - sqrt(delta) / (2 \* a));

x1 = (-b + sqrt(delta) / (2 \* a));

cout << "Ecuatie de gradul 2. Solutiile sunt " << x1 << " si " << x2;

}

}

}

return 0;

}

# Exercitiul 2

 Operatii pe numere intregi. Se  
 citesc doua numere intregi. In functie de operatia aleasa, se vor executa  
 diverse operatii pe acestea (adunare, diferenta, inmultire si impartire).

int main()

{

int a{ 0 }, b{ 0 };

float result{ 0.0 };

char option{0};

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

while (option != 'X' && option != 'x')

{

cout << "A. Adunare" << endl;

cout << "B. Scadere" << endl;

cout << "C. Inmultire" << endl;

cout << "D. Impartire" << endl;

cout << "X. Inchide programul" << endl;

cout << endl;

cout << "Alegeti o optiune: " << endl;

cin >> option;

if (option == 'a' || option == 'A')

{

result = a + b;

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

}

if (option == 'b' || option == 'B')

{

result = a - b;

cout << a << " - " << b << " = " << result << endl;

}

if (option == 'c' || option == 'C')

{

result = a \* b;

cout << a << " \* " << b << " = " << result << endl;

}

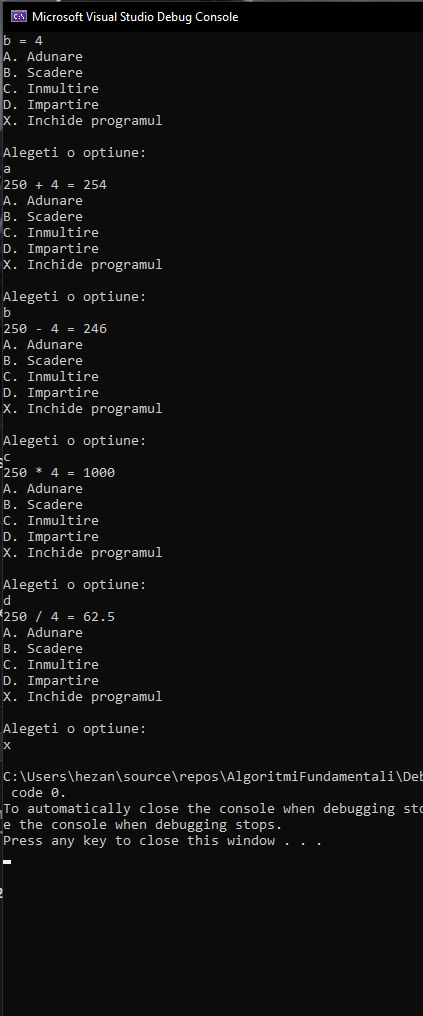
if (option == 'd' || option == 'C')

{

result = (float)a / b;

cout << a << " / " << b << " = " << result << endl;

}

 }

return 0;

}

# Exercitiul 3

**3.** Se va modifica aplicația cu ecuația de gradul al doilea, astfel încât se va permite  
 utilizatorului reluarea operațiunii, pentru execuții multiple (instructiunea de  
 salt **goto**).

int main()

{

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

float x1{ 0.0 }, x2{ 0.0 }, delta{ 0.0 };

char option{ 0 };

ecuatia\_2:

{

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

if (a == 0)

{

if (b == 0)

{

if (c == 0)

{

cout << "Ecuatie de gradul 0. Solutii infinite";

}

else

{

cout << "Ecuatie de gradul 0. Solutii multimea vida";

}

}

else

{

x1 = (float)-c / b;

cout << "Ecuatie de gradul 1. Solutia este:" << x1;

}

}

else

{

delta = b \* b - 4 \* a \* c;

if (delta < 0)

{

cout << "Ecuatie de gradul 2. Solutii din C";

}

else

{

if (delta == 0)

{

x1 = -b / (2 \* a);

cout << "Ecuatie de gradul 0. Solutie unica: " << x1;

}

else

{

x1 = (-b - sqrt(delta) / (2 \* a));

x1 = (-b + sqrt(delta) / (2 \* a));

cout << "Ecuatie de gradul 2. Solutiile sunt " << x1 << " si " << x2;

}

}

}

}

cout << endl;

cout << "Continuati operatiile? D (da) / N (nu)" << endl;

cin >> option;

if (option == 'd' || option == 'D')

{

goto ecuatia\_2;

}

else

{

cout << " :( ... ok ";

}

return 0;

}

# Exercitiul 4

**4.** Se  
 introduce de la tastatură un numar intreg **n**,  
 să se calculeze valoarea absoluta a sa, fără a se folosi o funcție predestinată.

int main()

{

int n{ 0 };

cout << "numarul este: ";

cin >> n;

if (n < 0)

{

cout << "Valoarea absoluta a numarului este: " << n \* -1;

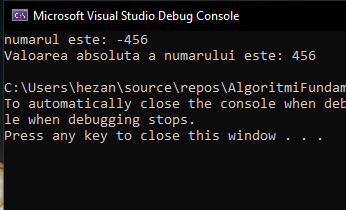
}

else

{

cout << "Valoarea absoluta a numarului este: " << n;

}



cout << endl;

return 0;

}

# Exercitiul 5

Se citesc 3 numere **a**, **b** și **c**. Să verifice daca pot fi termenii unei progresii aritmetice.

int main()

{

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

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

if (2 \* b == a + c)

{

cout << "Numere in progresie aritmetica";

}

else if (2 \* c == a + b)

{

cout << "Numere in progresie aritmetica";

}

else if (2 \* a == b + c)

{

cout << "Numere in progresie aritmetica";

}

else

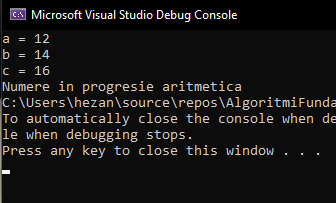
{

cout << "Numerele nu sunt progresie aritmetica";

}

return 0;

}



# Exercitiul 6

**6.** Se citesc 3 numere **a**, **b** și **c**. Aflati daca aceste numere pot fi laturile unui triunghi, de ce  
 tip e el. Se va transmite și aria calculată.

int main()

{

int a{ 0 }, b{ 0 }, c{ 0 }, ariaTriunghi{ 0 }, p{0};

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

if (a != 0 && b != 0 && c != 0)

{

if (a < b + c && b < a + c && c < a + b)

{

cout << "Laturie pot forma un triunghi." << endl;

p = (a + b + c) / 2;

ariaTriunghi = sqrt(p \* (p - a) \* (p - b) \* (p - c));

cout << "AriaABC = " << ariaTriunghi << endl;

}

else

{

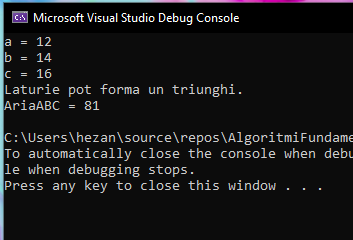
cout << "Laturie nu pot forma un triunghi." << endl;

}

}

return 0;

}



# Exercitiul 7

**7.** Se citesc trei numere reale. Sa se calculeze minimul si maximul valorilor  
 lor absolute.

int main()

{

int a{ 0 }, b{ 0 }, c{ 0 }, maximum{ 0 }, minimum{0};

vector <int> myVector(3,0);

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

myVector.at(0) = a;

myVector.at(1) = b;

myVector.at(2) = c;

minimum = myVector.at(0);

maximum = myVector.at(0);

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

{

if (myVector.at(i) <= minimum)

{

minimum = myVector.at(i);

if (myVector.at(i) < 0)

{

minimum = myVector.at(i) \* -1;

}

}

if (myVector.at(i) >= maximum)

{

maximum = myVector.at(i);

if (myVector.at(i) < 0)

{

maximum = myVector.at(i) \* -1;

}

}

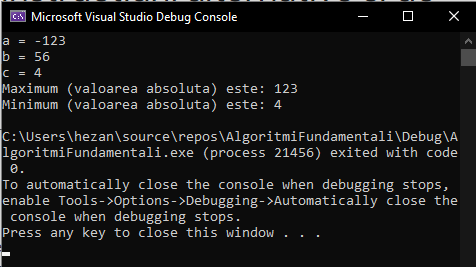
}

cout << "Maximum (valoarea absoluta) este: " << maximum << endl;

cout << "Minimum (valoarea absoluta) este: " << minimum << endl;

return 0;

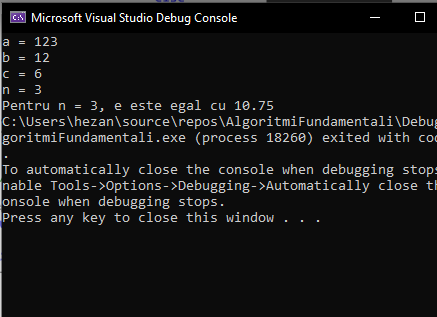
}



# Exercitiul 8

Se introduc de la tastatura **n**, **a**, **b**,  
**c**. Să se calculeze valoarea lui **e** definit astfel:  
            |(a+b)/c,  
ptr. n=1  
e=        |(b+c)/a,  
ptr. n=2  
            |(c+a)/b,  
ptr. n=3  
  
Se cere atât varianta folosind instrucțiuni alternative  
simple, cât și multiple

**Varianta 1**

int main()

{

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

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

cout << "n = ";

cin >> n;

if (n == 1)

{

e = (a + b) / (float)c;

cout << "Pentru n = 1, e este egal cu " << e;

}

else if (n == 2)

{

e = (b + c) / (float)a;

cout << "Pentru n = 2, e este egal cu " << e;

}

else if (n == 3)

{

e = (a + c) / (float)b;

cout << "Pentru n = 3, e este egal cu " << e;

}

else

{

cout << "Valorile introduse sunt invalide";

}

return 0;

}

**Varianta 2**

int main()

{

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

float e{};

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "c = ";

cin >> c;

cout << "n = ";

cin >> n;

switch (n)

{

case 1: e = (a + b) / (float)c; break;

case 2: e = (b + c) / (float)a; break;

case 3: e = (a + c) / (float)b; break;

default: cout << "Optiune invalida";

};

if (n == 1 || n == 2 || n == 3)

cout << "Optiunea aleasa este " << e << endl;

return 0;

}

# Exercitiul 9

Se citeste o literă de la tastatură. Să se determine dacă este vocală sau  
 consoană.

int main()

{

char option{};

cout << "Introduceti o litera : ";

cin >> option;

if (option == 'A' || option == 'a')

{

cout << "Caracterul introdus este o vocala";

}

else if (option == 'E' || option == 'e')

{

cout << "Caracterul introdus este o vocala";

}

else if (option == 'I' || option == 'i')

{

cout << "Caracterul introdus este o vocala";

}

else if (option == 'O' || option == 'o')

{

cout << "Caracterul introdus este o vocala";

}

else if (option == 'U' || option == 'u')

{

cout << "Caracterul introdus este o vocala";

}

else

{

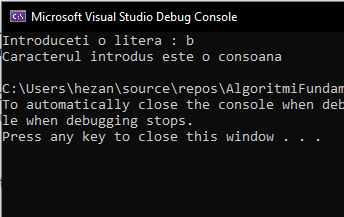
cout << "Caracterul introdus este o consoana";

}

cout << endl;

return 0;

}



# Exercitiul 10

Se dau  
 trei numere natural a,b,x. Să se scrie un algoritm care verifica  daca numarul apartine intervalului [a,b].  
 Dacă condiția este indeplinita  se va  
 return 1, altfel 0.

int main()

{

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

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "x = ";

cin >> x;

int aux{ 0 };

if (a > b)

{

aux = a;

a = b;

b = aux;

}

if (x >= a && x <= b)

{

cout << "1";

}

else

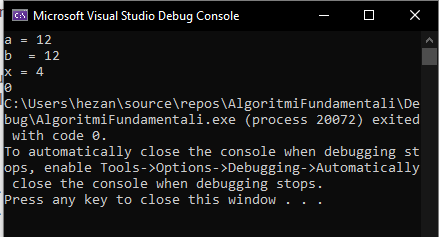
{

cout << "0";

}

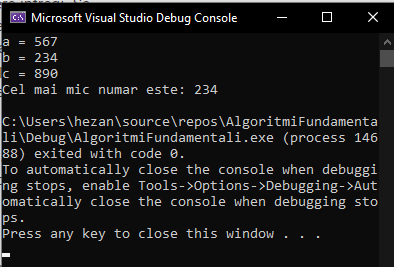
return 0;

}



# Exercitiul 11

Se dau  
 trei numere natural a,b,c. Sa se scrie un algoritm care determina si returneaza  
 minimul dintre cele trei numere.



int main()

{

vector <int> myVector(3, 0);

int minimum{};

cout << "a = ";

cin >> myVector.at(0);

cout << "b = ";

cin >> myVector.at(1);

cout << "c = ";

cin >> myVector.at(2);

minimum = myVector.at(0);

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

{

if (myVector.at(i) <= minimum)

{

minimum = myVector.at(i);

}

}

cout << "Cel mai mic numar este: " << minimum << endl;

return 0;

}