Nume si prenume: Paduraru Andra – Elena

Grupa: II A4

Tema: T0

**DESCRIERE ALGORITM:**

* Acest algoritm functioneaza pentru orice functie.
* In cazul de fata, pentru a gasi un punct maxim si unul minim aproximativ al functiei f(x1, x2) = (x[1] + 2 \* x[2] - 7) \* (x[1] + 2 \* x[2] - 7) + (2 \* x[1] + x[2] - 5) \* (2 \* x[1] + x[2] - 5) se utilizeaza o metoda euristica.
* In functia principala main() se va calcula minimul aproximativ si maximul aproximativ global al functiei. Prin intermediul functiei get\_random() se va alege aleatoriu un numar din codomeniul functiei utilizand o formula matematica.
* Primul pas este de a defini variabilele necesare si pentru un rezultat cat mai precis se va folosi un numar mare de rulari. Pentru a se afisa rezultate exacte se va utiliza "cout << fixed;" la inceputul functiei main().
* Al doilea pas este ca pentru fiecare numar i, care reprezinta a cata rulare este, sa se aleaga aleatoriu variabilele functiei (in acest caz sunt doua variabile) utilizandu-se functia get\_random(). Ulterior se calculeaza rezultatul functiei f(x1, x2) in functie de valorile lui x1 si x2.
* Al treilea pas este de a compara rezultatul functiei cu un minim si un maxim universal, ambele initializate la inceputul programului (daca rezultatul este cel mai mic gasit de pana acum, va fi retinut in minim, analog pentru maxim). Astfel se va retine minimul sau maximul din n rulari, precum si valorile variabilelor.
* Ultimul pas este de a afisa minimul si maximul gasit din cele n rulari pentru valoarea functiei si parametrii care au dat rezultatele respective.
* Algoritmul prezentat se va rezolva intr-un timp de rulare proportional cu n, unde n reprezinta variabila "rulari". => O(n).

**PSEUDOCOD PENTRU FUNCTIA ALEASA:**

random (double limita\_minima, double limita\_maxima)

{

return random\_intre\_limita\_minima\_si\_limita\_maxima;

}

int main ()

{

cout << fixed;

int rulari = 1000000, numar\_var = 2, i, j;

double limita\_minima = -10, limita\_maxima = 10, x[3], rezultat, min\_functie = 100000, max\_functie = -100000, xfmin[3], xfmax[3];

for (i = [0, rulari))

{

for (j = [1, numar\_var])

x[j] = random (limita\_minima, limita\_maxima);

rezultat = ecuatia\_functiei;

if (rezultat <= min\_functie)

{

min\_functie = rezultat;

for (j = [1, numar\_var])

{

xfmin[j] = x[j];

}

}

if (rezultat >= max\_functie)

{

max\_functie = rezultat;

for (j = [1, numar\_var])

{

xfmax[j] = x[j];

}

}

}

cout << min\_functie;

for (j = [1, numar\_var])

cout << xfmin[j];

cout << max\_functie;

for (j = [1, numar\_var])

cout << xfmax[j];

return 0;

}

**COD IN C++ PENTRU FUNCTIA ALEASA:**

#include <bits/stdc++.h>

using namespace std;

double get\_random(double limita\_minima, double limita\_maxima)

{

return limita\_minima + rand() / (RAND\_MAX / (limita\_maxima - limita\_minima));

}

int main()

{

cout << fixed;

int rulari = 1000000, numar\_var = 2, i, j;

double limita\_minima = -10, limita\_maxima = 10, x[3], rezultat, min\_functie = 1000000, max\_functie = -1000000, xfmin[3], xfmax[3];

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

{

for (j = 1; j <= numar\_var; j++)

x[j] = get\_random(limita\_minima, limita\_maxima);

rezultat = (x[1] + 2 \* x[2] - 7) \* (x[1] + 2 \* x[2] - 7) + (2 \* x[1] + x[2] - 5) \* (2 \* x[1] + x[2] - 5);

if (rezultat <= min\_functie)

{

min\_functie = rezultat;

cout << "Minimul functiei este: " << min\_functie <<'\n';

for (j = 1; j <= numar\_var; j++)

{

xfmin[j] = x[j];

cout<<"xfmin "<<j<<" este "<<xfmin[j]<<'\n';

}

cout << '\n';

}

if (rezultat >= max\_functie)

{

max\_functie = rezultat;

cout << "Maximul functiei este: " << max\_functie <<'\n';

for (j = 1; j <= numar\_var; j++)

{

xfmax[j] = x[j];

cout<<"xfmax "<<j<<" este "<<xfmax[j]<<'\n';

}

cout << '\n';

}

}

cout << "Minimul functiei: " << min\_functie << '\n';

for (j = 1; j <= numar\_var; j++)

cout << "x" << j << " = " << xfmin[j] << '\n';

cout << '\n';

cout << "Maximul functiei: " << max\_functie << '\n';

for (j = 1; j <= numar\_var; j++)

cout << "x" << j << " = " << xfmax[j] << '\n';

return 0;

}

**DETALII IMPLEMENTARE:**

Reprezentare: Numerele utilizate pentru functie si variabilele ei sunt declarate numere reale. Cele care reprezinta un numar fix (numarul de rulari, numarul variabilelor etc.) sunt declarate ca numere intregi.

Notiunea de vecinatate: Se aleg aleatoriu numerele pentru valorile functiei si parametrii ei.

Procedura de initializare: In afara de parametrii functiei si valoarea ei, toate variabilele sunt declarate la inceputul programului.

Conditia de oprire: Programul se va opri dupa un numar n dat de rulari.

Parametrii: Singurii parametri care apar in antetul unei functii sunt cei din functia get\_random() de tip “double”, ce au ca rol fixarea codomeniului si ajuta la alegerea unei valori aleatoare pentru x1 si x2.

**REZULTATE EXPERIMENTALE:**

Dupa n = 30 testari:

Timpul mediu: 0.417(90)

Timpul minim: 0.359

Timpul maxim: 1.070

Cea mai buna solutie = cea mai slaba solutie = media solutiilor:

xfmin = {3.000885, 0.990936}

xfmax = {-9.973754, -9.996338}

functie\_min = 0.000351

functie\_max = 2587.589290





