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

x1fmin	x2fmin	fmin	nr_rulari_min
-9.974975	1.271706	768.929250	1
-6.133915	6.174810	123.670211	2
1.700186	-0.402539	41.278379	3
4.210028	0.270699	18.677758	7
0.633259	1.423688	17.721078	14
2.035279	2.143315	1.933295	15
0.340281	3.259682	1.142782	67
0.602741	3.712577	1.063282	229
1.201514	2.480850	0.713695	546
0.773644	2.884304	0.532621	1251
0.890835	3.240761	0.139153	1513
1.065401	3.036287	0.046956	1925
0.924406	2.983184	0.040156	2810
0.948210	3.086337	0.014910	15877
1.050142	2.969146	0.004954	63621
0.962859	3.022858	0.002718	84904
0.990936	3.000885	0.000351	500069

x1fmax	x2fmax	fmax	nr_rulari_max
-9.974975	1.271706	768.929250	1
-3.920103	-9.700308	1427.420772	8
-7.618336	-9.906613	2094.150092	12
-9.967650	-8.779870	2328.853348	92
-9.598376	-9.860225	2478.934804	555
-9.938963	-9.718009	2520.017961	5647
-9.960326	-9.860225	2555.188733	8515
-9.970702	-9.923704	2571.148980	20092
-9.962767	-9.990234	2583.913467	135631
-9.967040	-9.990234	2584.826056	250779
-9.973754	-9.996338	2587.589320	925593