

```
1 // W4_1.cpp : Defines the entry point for the console application.
2 //
3 //          Wykład 4
4 //          Temat: Otwieranie i zamknięcie plików. Operacje petli
5
6 #include "stdafx.h"
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <math.h>
10
11
12 //definicja zmiennych globalnych
13 static FILE *pInp = NULL; //ta zmienna jest dostępna w dowolnym miejscu pliku bieżącego
14 static FILE *pOut = NULL; //ta zmienna jest dostępna w dowolnym miejscu pliku bieżącego
15
16 //deklaracja: funkcji są dostępne tylko w pliku bieżącym
17 static int my_fun(double x_start, double x_end, double d_x);
18 static void my_exception_print(char *str);
19 static void All_Free();
20
21
22 int main(int argc, char* argv[])
23 {
24     int retval = -1;
25     double x_st = 0.0, x_en = 0.0, inc_x = 0.0;
26
27     //Otwieranie pliku z danymi źródłowymi
28     pInp = fopen("input.txt", "r");
29     if(!pInp)
30     {
31         my_exception_print("Plik input.txt nie pozostał otwarty");
32     }
33
34     //Otwieranie pliku dla zapisu wyników
35     pOut = fopen("output.txt", "w");
36     if(!pOut)
37     {
38         my_exception_print("Plik output.txt nie pozostał otwarty");
39     }
40
41     //Czytaj dane z pliku input.txt
42     retval = fscanf(pInp, "%lf%lf%lf", &x_st, &x_en, &inc_x);
43     if(retval != 3)
44         my_exception_print("Błąd przy czytaniu danych z pliku input.txt");
45
46     //Oblicz tablice wartości funkcji
47     if(!my_fun(x_st, x_en, inc_x))
48     {
49         printf("Wprowadzone dane: x_st = %lf    x_en = %lf    inc_x = %lf\n",
50             x_st, x_en, inc_x);
51         my_exception_print("my_fun: błąd danych");
52     }
53     //zamkni pliki
```

```

54     All_Free();
55
56     system("notepad.exe output.txt");
57     //system("pause");
58
59     return 0;
60 }
61
62
63 int my_fun(double x_start, double x_end, double d_x)
64 /
65     *=====
66     =====
67     Obliczenie tablicy wartosci funkcji 1.0/(sqrt(1.0-x*x+5.0*x*x*x))
68     IN:  x_start - poczatkowe znaczenie argumentu x
69          x_end   - koncowe znaczenie
70          d_x     - przyrost argumenty (krok obliczenia)
71     OUT:
72     RetVal: 1 - sukces
73            0 - blad
74     =====
75     =====*/
76 {
77     double x; //biezaca wartosc argumentu
78     double f; //biezaca wartosc funkcji
79     double tmp;
80     const double eps = 1.0e-40;
81
82     if(x_start > x_end || !pOut || d_x < eps)
83         return 0;
84
85     fprintf(pOut, "\n\tTablicowanie funkcji\n\t poczatek: %lf \tkoniec: %lf \tkrok: %lf\n",
86             x_start, x_end, d_x);
87     fprintf(pOut, "\n\t x\t\t\t f(x)\n");
88
89     x = x_start;
90     while(x <= x_end)
91     {
92         tmp = 1.0-x*x+5.0*pow(x, 3.0);
93
94         if(tmp < eps )
95         {
96             x += d_x; //jest konieczne zeby uniknac nieskonczonej petli
97             continue; //Uwazamy, ze jesli to jest wartosc mala -
98                       //dzielenie przez 0 - obejosc ten punkt
99                       //lub wartosc ujemna - sqrt(...) problem - obejosc
100         }
101
102         f = sqrt(tmp);
103
104         f = 1.0/f;
105
106         //drukuj wynik. Uwaga: obiekt pOut jest zmienna globalna.
107         fprintf(pOut, "\t %e\t\t %e\n", x, f);

```

```
105
106     x += d_x;
107 }
108
109     return 1;
110 }
111
112
113 void All_Free()
114 /
115     *=====
116     =====
117     Zamykamy pliki
118     =====
119     =====*/
120 {
121     if(pInp)
122     {
123         fclose(pInp);
124         pInp = NULL;
125     }
126
127     if(pOut)
128     {
129         fclose(pOut);
130         pOut = NULL;
131     }
132 }
133
134 void my_exception_print(char *str)
135 {
136     printf("%s\n", str);
137     All_Free();
138     int ch;
139     while((ch = getchar()) != EOF)
140     ;
141     exit(1);
142 }
```