**Piotr Zawadka Środa, godzina 10-12**

**Projekt 1**

**1.Treść projektu**

Oblicz metodą prostokątów z niedomiarem całkę:

Użytkownik powinien mieć możliwość wprowadzenia dowolnej wartości określającej podział przedziału całkowania z i y oraz wartości a i b. Jednocześnie co 20% wykonanych obliczeń program powinien wyświetlić aktualny wynik.

**2.Propozycja rozwiązania**

1.Po wprowadzeniu odpowiednich współczynników **a** i **b** w postaci liczb rzeczywistych oraz przedziału całkowania oznaczonego <**z**,**y**> za pomocą radianów, program ma za zadanie sprawdzić, czy uda się zmniejszyć zakres całkowania o wielokrotność **2π/|b|** odpowiednio dla **z** i **y**.

2.Sprawdza warunek (**-y==z**) bądź (**y==z**). Gdy jest to prawda, wypisuje wynik całkowania równy zero, a następnie pyta się czy zakończyć program. W przeciwnym wypadku pyta się o dokładność z jaką użytkownik chce obliczyć wprowadzoną przez siebie całkę (**n** – dokładność wyrażana w ilości prostokątów, na którą dana całka ma zostać podzielona[liczba całkowita]).

3.Wypisuje funkcję całkowaną , wylicza deltę przesunięcia, a następnie wykonuje główną pętlę programu, która wylicza całkę, za pomocą metody prostokątów z niedomiarem.

4.Co 20% obliczeń wypisuje aktualny wynik tej całki w postaci [xE+000].

5.Po zakończeniu obliczeń pyta się czy zakończyć program.

**3.Zmienne**

**char** Koniec; -Zmienna kończenia programu

**double** a,b; -Współczynniki **a** i **b** funkcji

**double** z,y; -Przedział całkowania, gdzie **z** jest początkiem a **y** końcem

**double** dx; -Przesunięcie **x**

**double** calka; -Wynik całkowania

**double** wzor; -Wzór potrzebny gdy **f(x)** różne znakowo od **f(x+dx)**

**unsigned long** n; -Dokładność (ilość prostokątów)

**unsigned long** i; -Licznik

**int** p; -Współczynnik procentów (**p**\*20%)

**4.Stałe**

**#define** PI 3.14159265359 -Stała określająca liczbę Pi

**#define** F1 funkcja(y+i\*dx,a,b) -Makro, które odnosi się do funkcji „funkcja”; **f(x)**

**#define** F2 funkcja(y+(i+1)\*dx,a,b) -Makro, które odnosi się do funkcji „funkcja”; **f(x+dx)**

**5.Funkcje**

**void** dane(){};  
- Funkcja wypisująca do czego służy program, wraz z danymi autora.

**double** funkcja(**double** x, **double** a, **double** b){};  
- Funkcja obliczająca wartość funkcji całkowanej o zadanych współczynnikach **a** i **b** w punkcie **x.**

**char** end(**char** fin){};  
- Funkcja, która pyta się użytkownika czy chce zakończyć działanie programu.

**int** main(){};  
- Główna funkcja programu

6**.Testowanie**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Wpisywane dane** | | | | | **Rezultat** | **Wnioski** |
| **a** | **b** | **z** | **y** | **n** |
| 1 | 1 | 0 | 1 | 100 | 4.583907E-004 | Im większe **n** tym bardziej dokładny wynik |
| 1 | 1 | 0 | 1 | 100000000 | 4.568681E-004 |
| 1 | 1 | 0 | 1 | 1 | - | Nie udałoby się przedstawić wyników co 20%, a dokładność byłaby tragiczna |
| 1 | 1 | 0 | 1 | ULONG\_MAX | ERROR | Error, którego źródła dojść nie mogłem |
| 1 | 1 | 0 | 1 | 858993449 | 4.568681E-004 | Granica dokładności dla tego programu. Odnaleziona po kilkunastu próbach. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | - | - | - | - | 0 | Gdy **a** lub **b** jest 0 to nie ma potrzeby wpisywania dalszych danych, bo wynik to 0 |
| 1 | 0 | - | - | - | 0 |
| 1 | 1 | -1 | 1 | - | 0 | Gdy **z** i **y** są liczbami przeciwnymi lub są sobie równe, to wynik jest równy 0 |
| 1 | 1 | 1 | 1 | - | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1.79769E+308 | - | - | - | - | ERROR | Niezależnie co wpiszemy następnie, wyjdzie ERROR, gdyż współczynnik **a** jest za duży, tudzież za mały. |
| -1.79769E+308 | - | - | - | - | ERROR |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2.22507e-308 | 1 | 0 | 180 | 10000000 | 4.450139E-308 | Pomimo wyjścia poza zakres zmiennej **double**, wynik jest poprawny |
| -2.22507e-308 | 1 | 0 | 180 | 10000000 | -4.450139E-308 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1.79769E+308 | - | - | - | ERROR | Zbyt długi czas oczekiwania. Program nie odpowiada na komunikaty. |
| 1 | -1.79769E+308 | - | - | - | ERROR |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2.22507E-308 | 0 | 180 | 10000000 | 1.098028E-307 | Gdy **DBL\_MIN** jest współczynnikiem **b** to program działa poprawnie. |
| 1 | -2.22507E-308 | 0 | 180 | 10000000 | 1.098028E-307 |

**7.Zakres danych po testowaniu**

**char** Koniec; { (Koniec)ϵ{‘T’,’N’} } – Wybór ograniczony do ‘T’ lub ‘N’

**double** a,b; { aϵR,bϵR: aϵ[-2E+160,2E+160], bϵ[-2E+9,2E+9] }

**double** z,y; { zϵR,yϵR }

**double** dx; { (dx)ϵR }

**double** calka; { (calka)ϵR }

**double** wzor; { (wzor)ϵR }

**unsigned long** n; { nϵZ: nϵ[5,858993449] }

**unsigned long** i; { iϵZ: iϵ[1,n] } – Samoczynnie zwiększa się od 1 do n

**int** p; { pϵZ: pϵ[1,5] } – Samoczynnie zwiększa się od 1 do 5