МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 3

з дисципліни “Основи програмування”

тема “\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП-92  Мовчан Максим Олександрович  (*прізвище, ім’я, по батькові*)  варіант №9 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2019

**Мета роботи**

Навчитися створювати користувацькі функції із параметрами числових типів даних.  
Застосувати циклічні конструкції для інтрегрування математичних функцій.  
Навчитись програмно перевіряти ввід користувача і сповіщати його про можливі помилки вводу.

**Постановка завдання**

Використати математичну формулу з [попередньої лабораторної роботи](http://progbase.herokuapp.com/modules/progbase/tasks/lab2).  
Винести код обчислення значення y у нову функцію:

double fy(double x)

Програмно обчислити інтеграл функції методом лівих прямокутників на проміжку від деякого x\_min до x\_max із кроком інтегрування x\_step.  
Винести код обчислення інтегралу у функцію:

double int\_fy(double x\_min, double x\_max, double x\_step)

Вивести результат обчислення у термінал.

Користувач може задавати проміжок обчислення інтегралу та крок інтегрування у терміналі:

Якщо x\_min більше або рівне x\_max - виводити користувачу помилку невірного вводу проміжку.

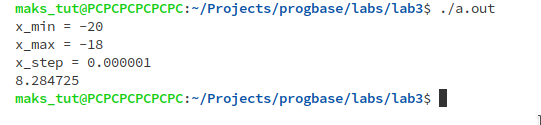
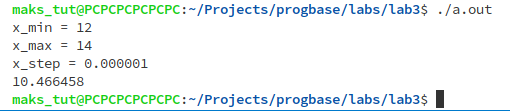
Якщо проміжок містить значення, що не входять у ОДЗ - виводити користувачу відповідну помилку.

ОДЗ: pi\*k +5; 5;

**Тексти коду програм**

|  |
| --- |
| **lab3.c** |
| #include <stdbool.h>  #include <math.h>  #include <stdio.h>  #include <stdlib.h>  # define M\_PI       3.14159265358979323846    double fy(double x) //Функция из lab2  {      double y;      if (x==5)      {          return 0;      }      else if (-5<x && x<=3)      {          y=(sqrt(x + 5) - 7);          return y;      }      else      {          y=(cos(x-5)/sin(x-5))-5;          return y;      }  }    double int\_fy(double x\_min, double x\_max, double x\_step) //Функция вычисления интеграла  {      double n = (x\_max-x\_min)/x\_step, sum; //n - кол-во шагов      for (double counter = x\_min; counter < x\_max; counter+=x\_step) //Вычисление функции для каждого значения счетчика (counter)      {          if (x\_min<=5 && x\_max>=5) //Если 5 в промежутке - значение функции 0          {              return 0;          }          sum+=fabs(fy(counter)); //сумма значений по модулю      }      return sum\*x\_step; //площадь  }    int check(double x\_min, double x\_max) //проверка на то, что значение pi+5 не принадлежит {xmin; xmax}  {      double limin = 5-33\*M\_PI;      double limax = 5+30\*M\_PI;      int z;      for(double count2=limin; count2 < limax; count2+=M\_PI) //перебор значений pi+5. если в промежутке - z и, соответственно, значение функции = 0      {          if (count2<x\_max && count2>x\_min)          {              z=0;              break;    //при нахождении pi+5 в {xmin; xmax} останавливает цикл          }          else          {              z=1;              continue;          }      }      return z;  }    int main()  {      double x\_min, x\_max, x\_step;      printf("x\_min = "); //Ввод      scanf("%lf", &x\_min);      printf("x\_max = ");      scanf("%lf", &x\_max);      if (x\_max<=x\_min) //min не >= max      {          printf("Min>=Max\n");          return 0;      }      else if(x\_min<5-33\*M\_PI || x\_max>5+30\*M\_PI)  //предупреждение при выходе за границы перебора      {          puts("\n!!!ANSWER MAY BE INCORRECT!!!\n");      }      if((x\_max<-5 || x\_min>3)&&check(x\_min, x\_max)==0)  //проверка на ОДЗ      {          puts("Error: ODZ(1)");          return 0;      }      else      {          printf("x\_step = "); //ввод xstep          scanf("%lf", &x\_step);          double S = int\_fy(x\_min, x\_max, x\_step); //интегрирование          if (S==0) //проверка на ОДЗ или S=0          {              puts("Error");              return 0;          }          else  //вывод S          {              printf("%.6lf\n", S);          }      }    } |

**Приклади результатів**

****

**Висновки**

Виконавши дану лабораторну роботу я навчитися створювати користувацькі функції із параметрами числових типів даних.  
Застосувати циклічні конструкції для інтрегрування математичних функцій.  
Навчитись програмно перевіряти ввід користувача і сповіщати його про можливі помилки вводу.