Бюджетное учреждение высшего образования   
Ханты-Мансийского автономного округа   
«Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 1 «Линейные алгоритмы»

по дисциплине «Программирование и основы алгоритмизации»

Выполнил: Хайитов Ш. Д.

студент группы 605-31

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2023 г.

Цель работы:

* Закрепление знаний о типах данных и операциях;
* Получение практических навыков постоения выражений, содержпщих арифмитические, логические и поразрядные операции;
* Получение практических навыков использования стандартных математических функций и функций стандартного ввода и вывода.

Задание:

Найти общее решение вычислительной задачи в соответствии с заданием, разработать алгоритм решения и реализовать программу.

Вариант 23. Заданы длительности двух интервалов времени и соответствующие скорости движения тела. Найти среднюю скорость движения тела.

Вариант 42. Заданы радиус окружности и количество сторон правильного многоугольника, вписанного в нее. Найти площадь многоугольника.

**Вариант 23**

**Формальное описание задачи.**

Длительность интервала движения это количество времени, которое двигалось тело (будем считаь в часах, а скорость в км\ч).

Средняя скорость – это вовсе не среднее арифметическое скоростей. По определению,



Для нахождения расстояния будем использовать формулу:



**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввести исходные данные – скорость первого и второго тела speed1 и speed2, промежуток движения тел time1 и time2.
2. Вычеслить сначало расстояние пройденные телами, а потом вычислить среднюю скорость.
3. Вывести результаты на экран.

**Блок-схема алгоритма программы.**

Начало

Ввод

*speed1, speed2, time1, time2*

Находим расстояния пройденное телами и скаладываем их

Складываем время, расстояния делим на время.

Вывод

*result*

Конец

**Листинг программы.**

/\* Лабораторная работа 1

Вариант 23. Заданы длительности двух интервалов времени и соответствующие скорости движения тела.

Найти среднюю скорость движения тела. \*/

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main() {

double speed1, time1, speed2, time2, result;

setlocale(LC\_ALL,"rus");

//Ввод данных

printf("Введите скорость первого тела(в км/ч): ");

scanf("%lf", &speed1);

printf("Введите промежуток времени движения первого тела(в часах): ");

scanf("%lf", &time1);

printf("Введите скорость второго тела(в км/ч): ");

scanf("%lf", &speed2);

printf("Введите промежуток времени движения второго тела(в часах): ");

scanf("%lf", &time2);

// Вычесления

result = ((speed1 \* time1) + (speed2 \* time2)) / (time1 + time2);

//Вывод результата

printf("Средняя скорость равна: %lf\n", result);

return 0;

system("pause");

}

**Пояснения к программе.**

В программе требуемые вычисления выполнены при помощи выражений:

result = ((speed1 \* time1) + (speed2 \* time2)) / (time1 + time2);

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h.

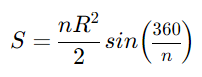
Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы).

**Вариант 42**

**Формальное описание задачи.**

Многоугольник находится внутри окружности, так как окружность описана

Для нахождения площади многоугольника можно использовать эту формаулу:



Так как функция sin() принимает только радианы:



**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввести исходные данные – количество сторон и радиус N и R соответственно.
2. Перевести 360 градусов в радиану
3. По формуле вычислить площадь
4. Вывести результаты на экран.

**Блок-схема алгоритма программы.**

Начало

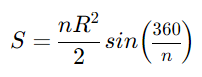
Ввод

N и S

Переводим 360 градусов в радианы по формуле:

rad = deg \* п / 180

Находим плащадь многоугольника по формуле:



Вывод

*result*

Конец

**Листинг программы.**

/\* Лабораторная работа 1

Вариант 42. Заданы радиус окружности и количество сторон правильного многоугольника, вписанного в нее. Найти площадь многоугольника. \*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

//task 2

int main() {

double N, R, S, rad;

setlocale(LC\_ALL,"rus");

//Ввод данных

printf("Введите радиус описанной окружности: ");

scanf("%lf", &R);

printf("Введите количество сторон правильного многоугольника: ");

scanf("%lf", &N);

// Вычесления

rad = 360 \* (M\_PI / 180); /\* Находим радиану \*/

S = (N \* pow(R, 2)) / 2 \* sin(rad / N); /\* Находим площадь \*/

//Вывод результата

printf("Площадь многоугольника равна: %lf\n", S);

system("pause");

return 0;

}

**Пояснения к программе.**

В программе требуемые вычисления выполнены при помощи выражений:

rad = 360 \* (M\_PI / 180); /\* Находим радиану \*/

А потом уже

S = (N \* pow(R, 2)) / 2 \* sin(rad / N); /\* Находим площадь \*/

Для возведения числа в квадрат использована математическая функция pow(x, n). Так как функция sin() считает только в радианах, мне пришлость сначало перевести 360 в радианы. Также использую M\_PI для числа пи . Для этого необходимо подключить заголовочный файл math.h.

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h.

Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы).

**Выводы:**

В задачах 23 и 42 для всех переменных выбран вещественный тип double, так как все величины могут быть не целыми числами, а тип double используется по умолчанию для вещественных вычислений, имеет достаточную точность и используется стандартными математическими функциями.

В задаче 42 необходимо было 360 градусов перевести в радианы, а потом уже подставлять в sin(), так как эта функция принимает значения только в радианах. Также был взят М\_PI вместо 3,14, что позволило получить более точные результаты.