

1. **Цель работы**

Изучение основной структуры линейной программы на языке Си.

1. **Постановка задачи**

Вариант 1: Из пункта А в пункт Б, выехала машина со скорость v1 км/ч. Навстречу ей выехал мотоциклист со скоростью v2 км/ч. На каком расстоянии находятся пункты А и Б, если машина и мотоциклист встретились через t часов?

1. **Формализация задачи**

Заданные скорости и время не могут быть отрицательны: v1<0, v2<0 и t<0. При нулевых скоростях не может быть ненулевое время встречи: v1==0, v2==0 и t!=0. Расстояние между пунктами А и Б – это сумма пройденных машиной и мотоциклом путей за указанное время: res = s1 + s2. Путь равен скорости, умноженной на время: s = v \* t.

1. **Набор тестовых примеров**

|  |  |  |  |
| --- | --- | --- | --- |
| Скорость машины | Скорость мотоцикла | Время, через которое они встретились | Расстояние между пунктами А и Б |
| 1 | 1 | 2 | 4 |
| 22.4 | 14.5 | 3 | 110.7 |

1. **Схема алгоритма**

Схема алгоритма программы представлена на рисунке 1.

Начало

v1=0, v2=0, t=0, s1=0, s2=0, res=0

Ввод v1, v2, t

Нет

Да

v1==0 && v2==0 && t!=0

Вывод «The values must not be negative! Please try again.»

Нет

Да

v1<0 || v2<0 || t<0

A

B

C

D

D

C

B

A

s1 = v1 \* t;

s2 = v2 \* t;

res = s1 + s2;

Вывод «The values must not be negative! Please try again.»

Вывод результата res

1 - Count again.

2 - Exit.

1

2

Конец

*Рисунок 1 – Схема алгоритма программы.*

1. **Листинг программы**

#if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#endif

#include <stdio.h>

int main(){

//defining local variables: speed of the car, speed of the bike and the time in which they met,

//distance for the car, distance for the bike and the result being the sum of the two.

float v1 = 0, v2 = 0, t = 0, s1 = 0, s2 = 0, res = 0;

int exit = 0; //exit flag from do while.

do{

printf("\nPlease, input the speed of the car:\t\t");

scanf("%f", &v1);

printf("Please, input the speed of the bike:\t\t");

scanf("%f", &v2);

printf("Please, input the time in which they met:\t");

scanf("%f", &t);

if (v1 < 0 || v2 < 0 || t < 0){ //checking if any of the variables are negative.

printf("The values must not be negative! Please try again. \n");

exit = 0;

}

else{

if (v1 == 0 && v2 == 0 && t != 0){ //checking if the variables are possible.

printf("The values are impossible! Please try again. \n");

exit = 0;

}

else{

s1 = v1 \* t; //calculating the distance for the car, for the bike and the result being the sum of the two.

s2 = v2 \* t;

res = s1 + s2;

printf("Marks A and B are located in %.2f kilometers away from each other.\n\n", res); //the output of the result.

printf("\nPlease choose from the following:\n\t1 - Count again.\n\t2 - Exit.\n"); //suggesting to user to run program again or exit.

scanf("%d", &exit);

switch (exit){

case 1: exit = 0; break;

case 2: exit = 1; break;

default: printf("Incorrect answer. The program will be closed."); exit = 1; break;

}

}

}

} while (exit != 1); // exit from cycle.

return 0;

}

1. **Выводы**

В ходе выполнения лабораторной работы была реализована программа, позволяющая вычислять расстояние между пунктом А и Б по заданным скоростям автомобиля и мотоцикла, а также времени, через которое они встретились.

Также, в ходе выполнения лабораторной работы было изучено, что главной функцией программы, написанной на языке Си, является функция main.

С помощью функции printf() осуществляется вывод на экран.

С помощью функции scanf() осуществляется ввод данных с клавиатуры.

В языке Си существуют следующие типы данных: int, float, double, char, short, long, unsigned, void.

Для ввода/вывода вещественных переменных следует применять следующие спецификации формата: %f, %lf.

Для ввода/вывода целочисленных переменных следует применять следующие спецификации формата: %d, %x, %o.

В языке Си поддерживаются следующие арифметические операции:

“\*” – умножение;

“/” – целочисленное деление;

“+” – сложение;

“-“ – вычитание;

“%” – остаток от целочисленного деления;

“++” – инкрементирование (увеличение на 1);

“--” – декрементирование (уменьшение на 1);

“-” – изменение знака.