

**Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
"Уфимский государственный авиационный технический университет"**

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Программирование

Отчет по лабораторной работе № _1_

Тема: «Управляющие структуры языка Си»

Группа ПМ-153	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Гайнетдинова А.А			

Уфа 2019

Цель: Ознакомиться с базовыми принципами построения программ и основными конструкциями языка C

Теоретический материал

==	Равно
!=	Не равно
>=	Больше либо равно
<=	Меньше либо равно
>	Больше
<	Меньше

Ответы на контрольные вопросы

1) Перечислите суффиксы, определяющие тип целой константы. Какой тип имеет целочисленная константа без суффикса?

В качестве суффиксов целочисленных констант используются: H, L. целочисленная константа без суффикса – integer.

2) Перечислите суффиксы, определяющие тип вещественной константы. Какой тип имеет вещественная константа без суффикса?

В качестве суффиксов вещественных констант используются: f, L, l. вещественная константа без суффикса – double.

3) Что такое «переменная»?

Переменная - поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным.

4) Объясните разницу между префиксной и постфиксной формами операций инкремента и декремента.

При использовании операции (инкремента\декремента) значение переменной, сначала, увеличивается на 1, если декремент уменьшается на 1, а затем используется в выражении. При использовании операции (инкремента\декремента) значение переменной сначала используется в выражении, а потом увеличивается на 1, если декремент уменьшается на 1.

5) К каким операндам применимы операции инкремента и декремента? К переменным со значениями.

6) Приведите примеры использования поразрядных операций и операций сдвига.

& (и) | (или) >>(сдвиг вправо) << (сдвиг влево)

7) Знаки каких бинарных операций могут использоваться в составных операциях присваивания?

Левым операндом операций (+=) (-=) может быть указатель.

8) Назовите обязательные этапы обработки, которые проходит исходная программа, подготовленная на языке Си в виде текстового файла.

3 этапа:

1. препроцессорное преобразование текста;
2. компиляция;
3. компоновка (редактирование связей или сборка).

9) Как обозначается пустой оператор?

Пустой оператор состоит только из точки с запятой.

10) Перечислите операторы цикла языка Си.

for, while, do..while

11) Перечислите операторы ветвления языка Си.

if, else

12) В чем сходства и различия операторов *break* и *continue*?

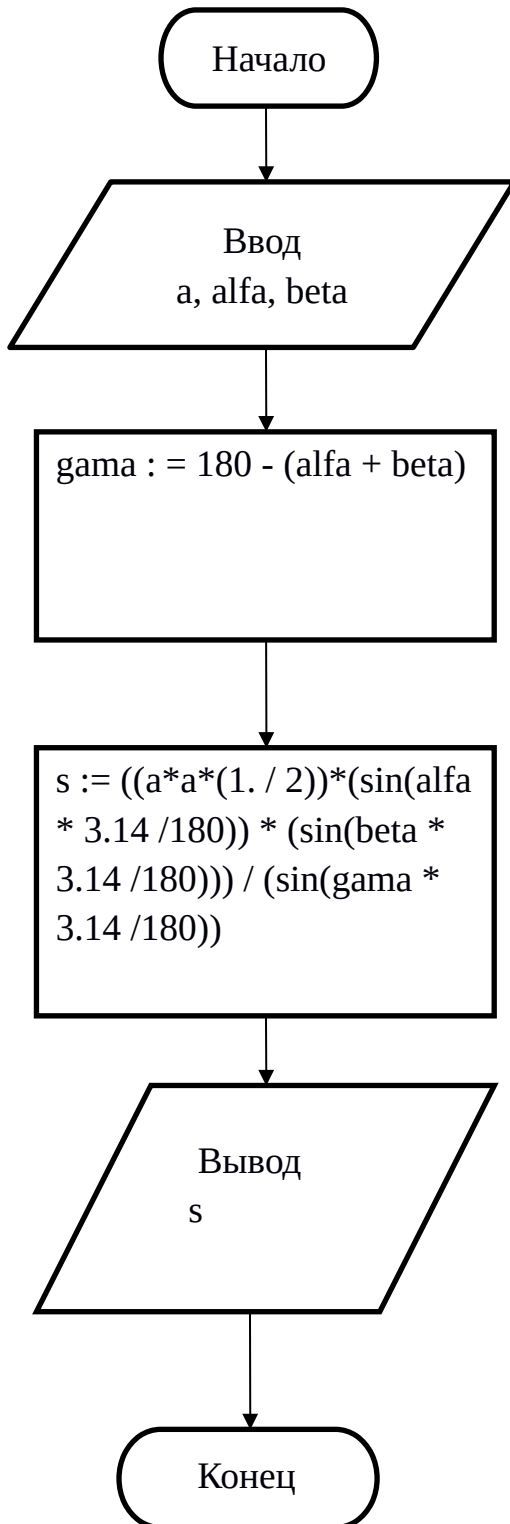
break вызывает немедленный выход из циклов, организуемых с помощью операторов for, while, do-while, а также прекращение оператора switch.

Оператор *continue* тоже предназначен для прерывания циклического процесса, организуемого операторами for, while, do-while. Но в отличие от оператора *break*, он не прекращает дальнейшее выполнение цикла, а только немедленно переходит к следующей итерации того цикла, в теле которого он оказался.

Индивидуальное задание №1

Задание: Написать программу вычисления площади треугольника, если известны длина одной из его сторон и прилежащие к ней углы. Ввести соответствующие переменные, проинициализировать их некоторыми значениями, исходные данные и результат расчета вывести на экран. Предусмотреть тестовый пример, который легко решается в уме.

Блок-схема



Описание программы:

a – известная сторона.

alfa – первый известный угол.

beta – второй известный угол.

s – искомая площадь.

gama – угол, который мы находим вследствие вычислений (третий угол)

Сперва находим неизвестный угол треугольника, где остальные 2 угла известны, затем по формуле находим площадь треугольника, чего сторону и 3 угла.

Исходный код программы

```
#include <math.h>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    float a, alfa, beta, s, gama;
```

```
    scanf("%f %f %f", &a, &alfa, &beta); //ввод значений стороны, первого  
угла и второго угла
```

```
    gama = 180 - (alfa + beta); //ищем неизвестный угол в треугольнике, где  
2 остальных угла известны
```

```
    s=((a*a*(1. / 2))*(sin(alfa * 3.14 /180)) * (sin(beta * 3.14 /180))) /  
(sin(gama * 3.14 /180));// находим искомую площадь треугольника, зная  
сторону и 3 угла треугольника
```

```
    printf("%f", s); //выводим искомую площадь.
```

```
    return 0;
```

```
}
```

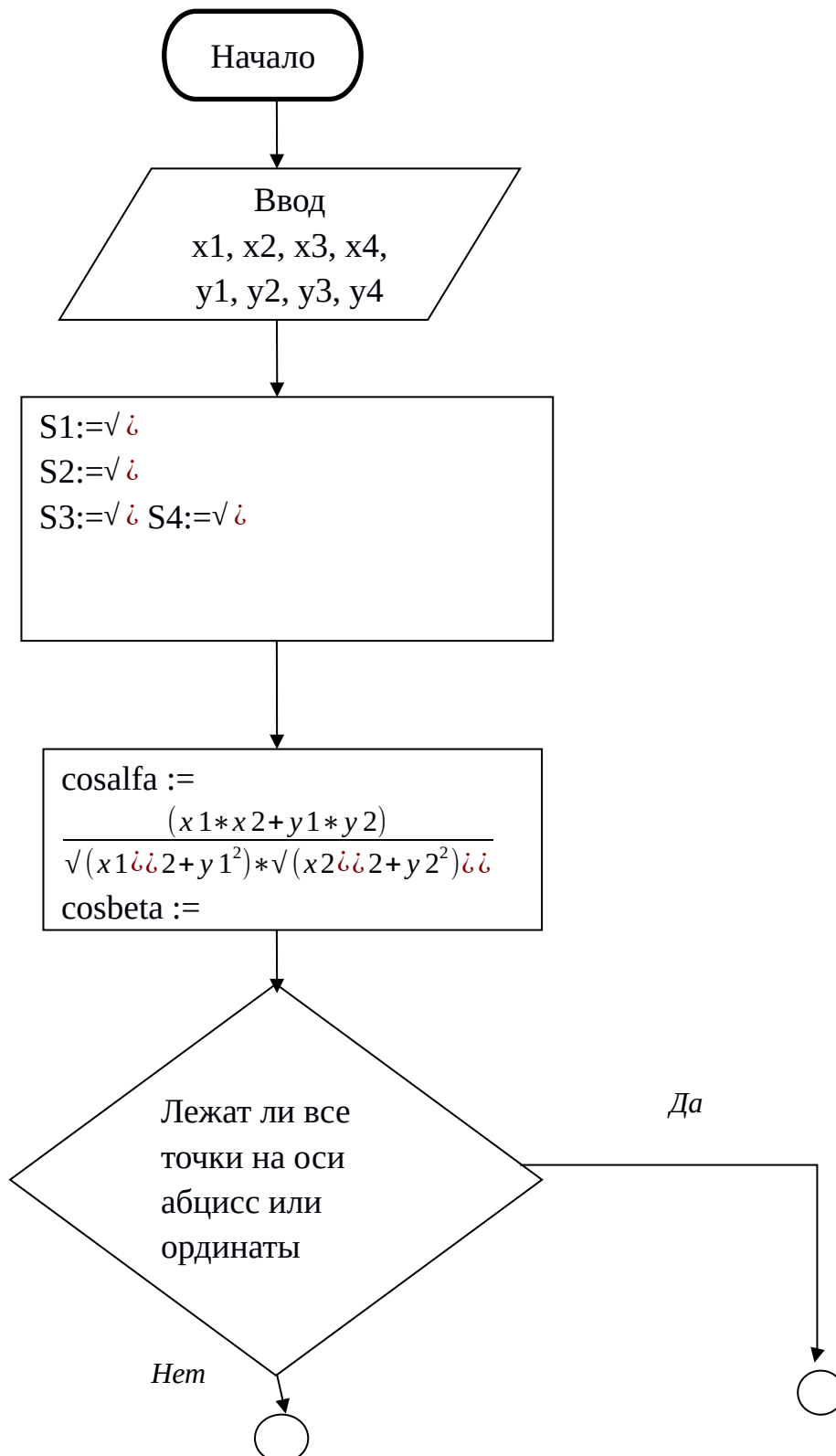
Пример выполнения программы

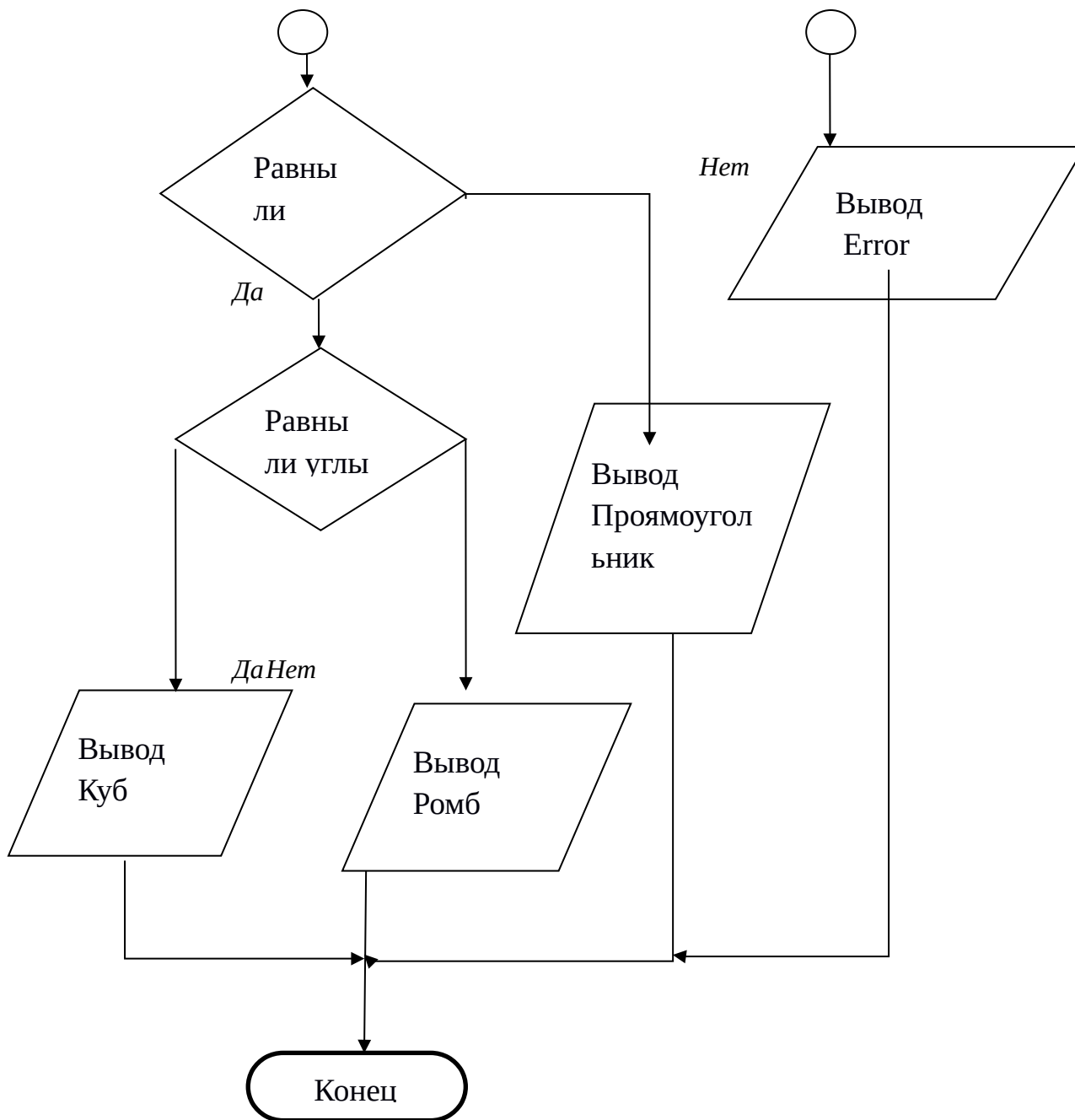
```
10  
60  
30  
21.634050  
Process returned 0 (0x0)   execution time : 3.849 s  
Press any key to continue.
```

Индивидуальное задание №2

Задание: Написать программу которая по координатам четырех точек на плоскости будет определять тип четырехугольника: ромб, прямоугольник, квадрат. Предусмотреть проверку некорректности исходных данных (например, все точки лежат на одной прямой).

Блок-схема





Описание программы:

$x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ – координаты 4 точек на плоскости.

S_1, S_2, S_3, S_4 – длины отрезков, соединяющие эти точки, то есть стороны четырёхугольника.

$\cos\alpha, \cos\beta$ – углы прилежащие одной прямой.

Сперва находим длины четырёхугольника.

Находим косинусы прилежащие одной стороне.

Если стороны равны – то четырёхугольник либо квадрат, либо ромб.

Находим косинусы и определяем вид четырёхугольника (квадрат или ромб)

Если стороны не равны – то это прямоугольник.

Исходный код программы

```
#include <math.h>
#include <stdio.h>
using namespace std;

int main()
{
    int x1, x2, x3, x4, y1, y2, y3, y4, S1, S2, S3, S4;
    float cosalfa, cosbeta;
    scanf("%d %d %d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3,
&x4, &y4); //вводим координаты
    S1=sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)); //находим сторону 1
    S2=sqrt((x3-x2)*(x3-x2) + (y3-y2)*(y3-y2)); //находим сторону 2
    S3=sqrt((x4-x3)*(x4-x3) + (y4-y3)*(y4-y3)); //находим сторону 3
    S4=sqrt((x1-x4)*(x1-x4) + (y1-y4)*(y1-y4)); //находим сторону 4

    cosalfa = (x1*x2 + y1*y2)/(sqrt(x1*x1 + y1*y1)* sqrt(x2*x2 +
y2*y2)); //находим угол 1, принадлежащий одной стороне
    cosbeta = (x2*x3 + y2*y3)/(sqrt(x2*x2 + y2*y2) * sqrt(x3*x3+y3*y3));
//находим угол 2, принадлежащий одной стороне

    if((x1 == x2) && (x2 == x3) && (x3 == x4) && (x4 == x1)){
        printf("Error"); //проверка, что точки не лежат на одной прямой
        return 0;
    }
    if((y1 == y2) && (y2 == y3) && (y3 == y4) && (y4 == y1)){
        printf("Error"); //проверка, что точки не лежат на одной прямой
        return 0;
    }
    if((S1 == S2) && (S2 == S3) && (S3 == S4) && (S4 == S1)){
        if(cosalfa == cosbeta){
            printf("KUB");//если стороны равны, и углы совпадают- это куб
        }
        else
            printf("Rhombus");//если стороны равны, а углы не совпадают-
это ромб
    }
    else
```



```
printf("Rectangle"); //если стороны не равны, и углы не совпадают-  
это прямоугольник  
return 0;  
}
```

Пример выполнения программы

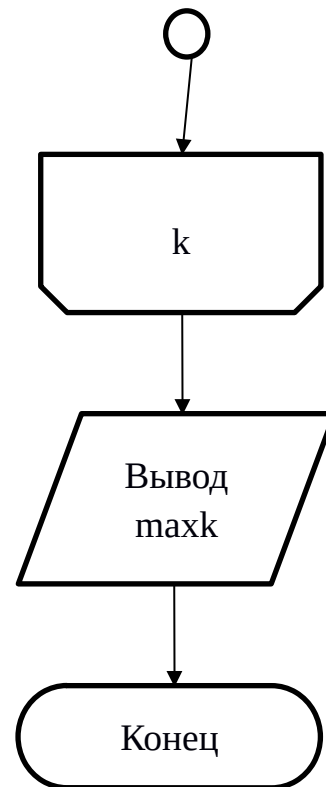
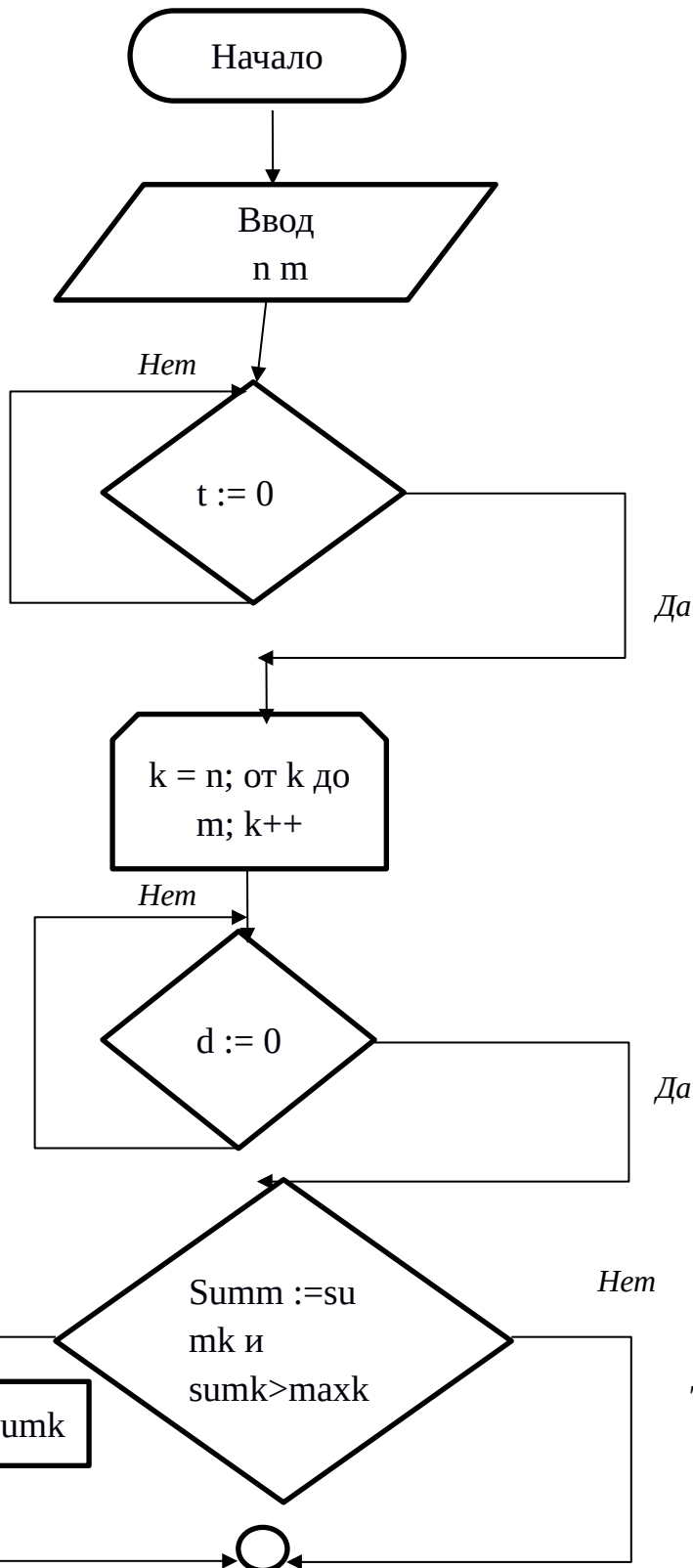
```
1  
1  
1  
1  
2  
2  
2  
2  
2  
2  
1  
KUB  
Process returned 0 (0x0)   execution time : 5.920 s  
Press any key to continue.
```

```
1  
2  
1  
4  
1  
3  
1  
6  
Error  
Process returned 0 (0x0)   execution time : 4.666 s  
Press any key to continue.
```

Индивидуальное задание №3

Задание: Задаются t и n . Определить наибольшее число k между t и n с суммой цифр как у числа t . Результат расчета вывести на экран. Предусмотреть тестовый пример, который легко решается в уме.

Блок-схема



Описание программы:

summ – сумма цифр в числе m.

sumk - сумма цифр в числе k.

maxk – максимальное число k.

t – переменная, равная по значению числу m.

d - переменная, равная по значению числу k.

Вводим числа n и m.

Находим сумму цифр в числе m.

Находим искомое k, последовательно проверяя каждое число от n до m;

Проверяем, чтобы оно было максимально.

Исходный код программы

```
#include <math.h>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int m, n, k, d, t, summ=0, sumk=0, maxk=0;
```

```
    scanf("%d %d", &n, &m); //Вводим n и m
```

```
    t=m;
```

```
    while(t!=0){ //находим сумму цифр в числе m
```

```
        summ +=t %10;
```

```
        t/=10;
```

```
    }
```

```
    for(k=n ;k<m ; k++){ //проверяем подходит ли каждое k под условие
```

```
        d=k;
```

```
        while(d!=0){
```

```
            sumk += d%10; //находим сумму цифр в текущем k
```

```
            d/=10;
```

```
        }
```

```
        if((summ == sumk)&&(sumk>maxk)){ //записываем максимальное,
```

удовлетворяющее условию

```
            maxk = k;
```

```
        }
```

```
        sumk=0;
```

```
    }
```

```
printf("%d", maxk); //выводим максимальное k, сумма цифр которого,  
равно сумме цифр m
```

```
return 0;  
}
```

Пример выполнения программы

```
11  
22  
13  
Process returned 0 (0x0)   execution time : 2.998 s  
Press any key to continue.
```

Вывод

Я научился основам языка Си, разобрал циклы и операторы условия, а так же счетчик.