МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 2

**Решение задач с использованием ветвящихся алгоритмов. Операции отношения, логические операции**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Осипенко

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

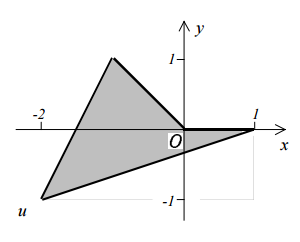
Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2020

1. **Формулировка задачи**

Определить, попадает ли точка с заданными координатами x и y в закрашенную область 

1. **Постановка задачи**

На вход программы поступают два вещественных числа в диапазоне +/- 1.7E-308, являющиеся координатами (x, y) точки на плоскости. По этим координатам проверяются условия принадлежности/непринадлежности данной точки частям плоской фигуры, принадлежащей плоскости. На выход поступает письменное сообщение о том, попадает или не попадает точка в заштрихованную область.

1. **Математическая модель**

Дан четырехугольник. Мы можем разбить его на четыре прямые: {y1=3(x+1.5); x in [-2;-1]}; {y2=-1x; x in [-1;0]}; {y3=0; x in [0;1]}; {y4 = (x-1)/3; x in [-2;1]}. Если точка попадает в закрашенную область, то ее координата y должна быть меньше(или равна) значению прямой, в область определения которой попадает ее координата x(верно для прямых y1,y2,y3), и быть больше( или равна) значению четвертой прямой(y4) в данной координате x.

1. **Описание алгоритма**

Начало

* 1. Ввод координат y и x
  2. Проверка условия : ((y1 >= y && x <= -1) or

(y2 >= y && (-1 <= x && x <= 0)) or

(y3 >= y && (0 <= x && x <= 1))) and

(y4 <= y && (-2.0 <=x && x <= 1.0)).

Если условие не выполняется то переход к пункту 4.4.

* 1. Вывод слова – Yes

Завершение работу программы

* 1. Вывод слова - No

Конец

1. **Опорный граф (блок-схема) алгоритма**

Начало

Ввод координат

((y1 >= y && x <= -1) or

(y2 >= y && (-1 <= x && x <= 0)) or (y3 >= y && (0 <= x && x <= 1))) and (y4 <= y && (-2.0 <=x && x <= 1.0))

False True

No

Yes

Конец

1. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

В данном случае программа состоит из блока ввода, блока условия и двух блоков вывода. Структуры данных – нет.

1. **Текст программы:**

**на языке С++**

#include <iostream>

using namespace std;

int main(){

double x,y;

// Вводим координаты x, y

cout << "Enter x: "; cin >> x;

cout << "Enter y: "; cin >> y;

//Определяем уравнения прямых

double y1 = 3\*(x+1.5),

y2 = -x,

y3 = 0,

y4 = (x-1)/3;

// Делаем проверку попадания точки в закрашенную область

bool isY1 = (y1 >= y && x <= -1),

isY2 = (y2 >= y && (-1 <= x && x <= 0)),

isY3 = (y3 >= y && (0 <= x && x <= 1)),

isY4 = (y4 <= y && (-2.0 <=x && x <= 1.0));

//Выводим результат

if ((isY1 || isY2 || isY3) && isY4)

cout << "Yes" << endl;

else cout << "No" << endl;

return 0;

}

**на языке Pascal**

var

y,x,y1,y2,y3,y4: real;

isY1, isY2, isY3, isY4: boolean;

begin

// Вводим координаты x, y

writeln('Enter x and y: ');

read(x,y);

//Определяем уравнения прямых

y1 := 3\*(x+1.5);

y2 := -x;

y3 := 0;

y4 := (x-1)/3;

// Делаем проверку попадания точки в закрашенную область

isY1 := ((y1 >= y) and (x <= -1));

isY2 := ((y2 >= y) and ((-1 <= x) and (x <= 0)));

isY3 := ((y3 >= y) and ((0 <= x) and (x <= 1)));

isY4 := ((y4 <= y) and ((-2.0 <=x) and (x <= 1.0)));

//Выводим результат

if ((isY1 or isY2 or isY3) and isY4) then

writeln('Yes')

else writeln('No')

end.

1. **Проверка работоспособности (тестирование) программы**

Проведем три теста:

x = -2, y = 0, программа выдала No; x = -0.5, y = 0.5, Yes; x = 0, y = 0, Yes.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программ различий в скорости работы нет или оно минимально

1. **Формулировка задачи**

Дано четырехзначное число. Определить: а) равна ли сумма двух первых его цифр сумме двух его последних цифр; б) кратна ли трем сумма его цифр; в) кратно ли четырем произведение его цифр; г) кратно ли произведение его цифр числу a.

1. **Постановка задачи**

На вход программы поступают два целых числа в диапазоне +/- 2 147 483 647, одно из которых является четырехзначным числом, другое числом а. Над четырехзначным числом проводятся операции, описанные в условии задачи. На выход поступает письменное сообщение для каждого пункта о том, выполняются ли вышеуказанные условия.

1. **Математическая модель**

Дано числа, мы должны разбить его на разряды. Например дано число 1234, чтобы получить четвертую цифру нужно поделить исходное на 10 и взять остаток: 4 = 1234 mod 10, затем 1234 / 10 = 123. Для остальных цифр тоже самое: 3 = 123 mod 10, 123 / 10 = 12; 2 = 12 mod 10, 12 / 10 = 1; 1 = 1 mod 10. Для пункта А складываем первую и вторую цифры и сравниваем с суммой третьей и четвертой цифры: (1 + 2) < (3 + 4), суммы не равны. Для пункта Б складываем цифры числа и делим результат на 3: (1 + 2 + 3 +4 ) / 3 = 3 и 1 в остатке, значит не суммы цифр числа не кратна 3. В пунктах В и Г перемножаем цифры числа друг с другом и делим на 4(а = 2): (1\*2\*3\*4)/4(а = 2) = 6(12), значит произведение цифр числа кратно 4(а = 2).

1. **Описание алгоритма**

Начало

* 1. Ввод чисел num и a
  2. Делим num на 10, результат записываем в num, остаток запоминаем в frst
  3. Делим num на 10, результат записываем в num, остаток запоминаем в scnd
  4. Делим num на 10, результат записываем в frth, остаток запоминаем в thrd
  5. Проверка условия: frst + scnd == thrd + frth

Если не удовлетворяет условие, вывести “ a)Sum of first and second numbers != sum of third and fourth “, перейти к пункту 4.7

* 1. Вывод “ a)Sum of first and second numbers = sum of third and fourth”
  2. Проверка условия: (frst + scnd + thrd + frth) % 3 == 0

Если не удовлетворяет условие, вывести “ b)Sum of first, second, third and fourth numbers aren't divided into 3“ , перейти к пункту 4.9

* 1. Вывод “b)Sum of first, second, third and fourth numbers are divided into 3”
  2. Проверка условия: frst \* scnd \* thrd \* frth % 4 == 0

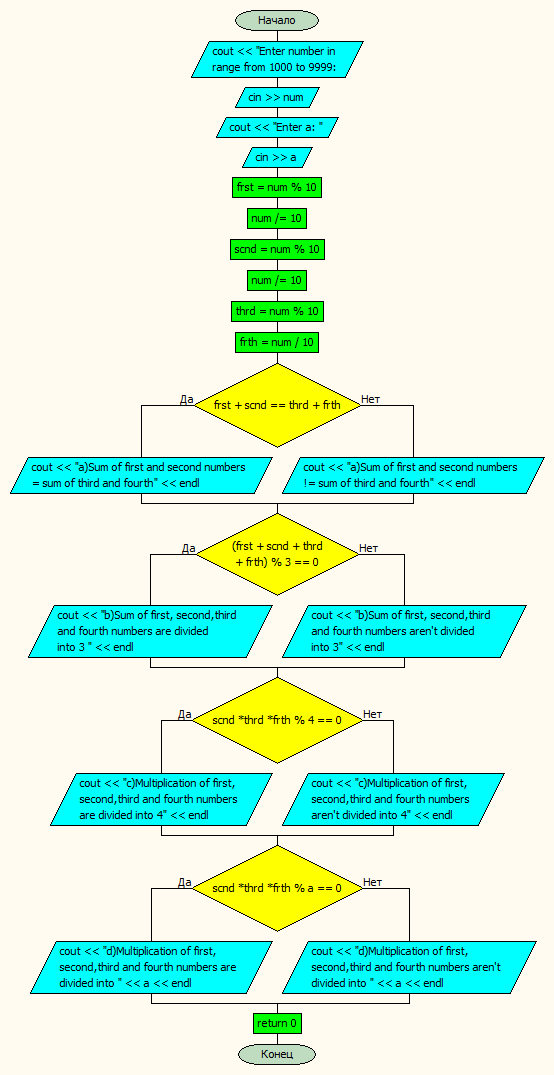
Если не удовлетворяет условие, вывести “c)Multiplication of first, second, third and fourth numbers aren't divided into 4“ , перейти к пункту 4.11

* 1. Вывод “c)Multiplication of first, second, third and fourth numbers are divided into 4”
  2. Проверка условия: frst \* scnd \* thrd \* frth % a == 0

Если не удовлетворяет условие, вывести “d)Multiplication of first, second, third and fourth numbers aren't divided into a“, завершить работу программы

* 1. Вывод “d)Multiplication of first, second, third and fourth numbers are divided into a”

Конец

1. **Опорный граф (блок-схема) алгоритма**
2. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

Программа состоит из четырех блоков условий, двух блоков ввода и десяти блоков вывода. Структура данных – нет.

1. **Текст программы:**

**на языке С++**

#include <iostream>

using namespace std;

int main()

{

int num, frst,scnd,thrd,frth, a;

//Ввод четырехзначного числа и а

cout << "Enter number in range from 1000 to 9999: "; cin >> num;

cout << "Enter a: "; cin >> a;

frst = num % 10;

num /= 10;

scnd = num % 10;

num /= 10;

thrd = num % 10;

frth = num / 10;

if(frst + scnd == thrd + frth)// Пункт а

cout <<"a)Sum of first and second numbers = sum of third and fourth" << endl;

else

cout <<"a)Sum of first and second numbers != sum of third and fourth"<< endl;

if((frst + scnd + thrd + frth) % 3 == 0)//Пункт б

cout <<"b)Sum of first, second,third and fourth numbers are divided into 3 "<< endl;

else

cout <<"b)Sum of first, second,third and fourth numbers aren't divided into 3"<< endl;

if(frst \* scnd \* thrd \* frth % 4 == 0)//Пункт в

cout <<"c)Multiplication of first, second,third and fourth numbers are divided into 4"<< endl;

else

cout <<"c)Multiplication of first, second,third and fourth numbers aren't divided into 4"<< endl;

if(frst \* scnd \* thrd \* frth % a == 0)//Пункт г

cout <<"d)Multiplication of first, second,third and fourth numbers are divided into "<< a << endl;

else

cout <<"d)Multiplication of first, second,third and fourth numbers aren't divided into "<< a<< endl;

return 0;

}

**на языке Pascal**

var

num, frst,scnd,thrd,frth, a:integer;

begin

//Ввод числа

writeln('Enter number in range from 1000 to 9999: ');

readln(num);

writeln('Enter a: ');

readln(a);

frst:= num mod 10;

num := num div 10;

scnd := num mod 10;

num := num div 10;

thrd := num mod 10;

frth := num div 10;

if frst + scnd = thrd + frth then // Пункт а

writeln('a) a)Sum of first and second numbers = sum of third and fourth ')

else

writeln('a) Sum of first and second numbers != sum of third and fourth ');

if (frst + scnd + thrd + frth) mod 3 = 0 then//Пункт б

writeln('b) Sum of first, second,third and fourth numbers are divided into 3 ')

else

writeln('b) Multiplication of first, second,third and fourth numbers arent divided into 3 ');

if frst \* scnd \* thrd \* frth mod 4 = 0 then//Пункт в

writeln('c) Multiplication of first, second,third and fourth numbers are divided into 4 ')

else

writeln('c) Multiplication of first, second,third and fourth numbers arent divided into 4 ');

if frst \* scnd \* thrd \* frth mod a = 0 then//Пункт г

writeln('d) Multiplication of first, second,third and fourth numbers are divided into ',a)

else

writeln('d) Multiplication of first, second,third and fourth numbers arent divided into ',a);

end.

1. **Проверка работоспособности (тестирование) программы**

(Для а = 2)Введем 1234: а)-, б)-,в)+,г)+; 7683: а)-, б)-,в)+,г)+;8383: а)+, б)-,в)+,г)+.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программ различий в скорости работы нет или оно минимально

Контрольные вопросы:

1. В чем отличие полной и сокращенной форм условного оператора?

- Полная форма позволяет учитывать и выполнять определенные операции не только, когда выражение истинно(блок if), но и когда ложны(блок else), в отличии от сокращенное формы, где есть только блок if.

1. Что может быть использовано в качестве выражения в условном операторе?

- Любой стандартный тип данных и структура, у которых предусмотрены операторы или методы, возвращающие булевы значения(или 0,отличное от 0).

1. Что называют константным выражением?

- Это выражения, членами которого являются константы, процесс вычисления происходит во-время компиляции(например лог. выражения, const double pi = 22.0/7.0 , …), и результатом является константа.

1. В чем отличие использования оператора if в ЯП Pascal и C++?

- В c++, в отличии от pascal, многие операторы должны иметь () для записи в них выражений[ if(expression) в c++ против if expression then в pascal ], также версии в pascal необходимо ключевое слово “then”.