МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра информатики и вычислительной техники (ИВТ)

Лабораторная работа №01

**Интегрированные среды разработки программного обеспечения**

Выполнил студент 585 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Губченко

Проверил: к.т.н,, доцент каф. ВТиЭ

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

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

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

Барнаул 2018

1. **Формулировка задачи**

Задано четырёхзначное число, определить сумму крайних цифр.

1. **Постановка задачи**

На вход поступает целое положительное 4-ёхзначное число. Программа выполняет разложение исходного числа на разряды и определяет сумму его крайних цифр. На выход поступает целое положительное число в диапазоне от 1 до 18.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Рассчитаем сумму крайних цифр числа 1234. Определим остаток от деления числа 1234 на 10, получаем 4. Далее прибегнем к целочисленному делению на 1000, получим цифру 1. Далее рассчитаем сумму цифр 1 и 4. Получим число 5.

1. **Описание алгоритма**

**Начало алгоритма**

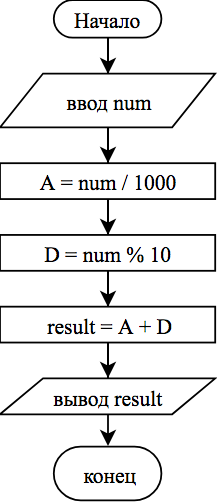
* 1. Ввод числа num
  2. Присвоение переменной D значения num mod 10
  3. Присвоение переменной A значения num div 1000
  4. Присвоение переменной result значения суммы A и D
  5. Вывод result

**Конец алгоритма**

1. **Проект программы с определением суммы крайних цифр четырёхзначного числа**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из трёх частей: 1 блок ввода, 1 блок арифметических вычислений и 1 блок вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, первой и четвёртой цифры, результата.

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



1. **Текст программы:**

**на языке С++**

*#include <iostream>*

*#include <cstdlib>*

*using namespace std;*

*int main (int argc, char const \*argv[]) {*

*int D;*

*int A;*

*int result;*

*int num;*

*cout << "enter the number" <<endl;*

*cin >> num;*

*A = num / 1000;*

*D = num % 10;*

*result = A + D;*

*cout << result << endl;*

*return 0;*

*}*

**на языке Pascal**

*{ Программа, которая определяет, сумму крайних цифр четырёхзначного числа }*

Program lab\_01;

var A, D, num, result: integer;

Begin

readln(num);

D:= num mod 10;

A:= num div 1000;

result:= A + D

writeln(result);

readln;

end;

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

Для проверки работоспособности нужно просто запустить программу.

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

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.

1. **Формулировка задачи**

Дано число a. Не пользуясь никакими арифметическими операциями кроме умножения получите

а) а4 за две операции

б) а6 за три операции

в) а15 за пять операций

1. **Постановка задачи**

На вход поступает вещественное число a. Программа выполняет умножение данного числа само на себя так, чтобы получились числа a4 за две операции, a6 за три операции, a15 за пять операций. На выход поступает три вещественных числа, определяющих значения выражений a4, a6, a15.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Рассчитаем степени числа 2.

22= 2\*2, 24=22\*22,26=24\*22, 23=22\*2, 26=23\*23, 212=26\*26,215=212\*23. Получим 24, 26, 215

1. **Описание алгоритма**

**Начало алгоритма**

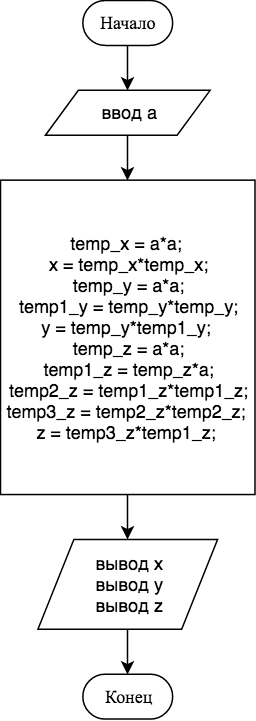
* 1. ввод числа a
  2. присвоение переменной temp\_x значения а\*а
  3. присвоение переменной x значения temp\_x\*temp\_x
  4. присвоение переменной temp\_y значения a\*a
  5. присвоение переменной temp1\_y значения temp\_y\*temp\_y
  6. присвоение переменной y значения temp\_y\*temp1\_y
  7. присвоение переменной temp\_z значения a\*a
  8. присвоение переменной temp1\_z значения temp\_z\*a
  9. присвоение переменной temp2\_z значения temp1\_z\*temp1\_z
  10. присвоение переменной temp3\_z значения temp2\_z\*temp2\_z
  11. присвоение переменной z значения temp3\_z\*temp1\_z
  12. вывод x
  13. вывод y
  14. вывод z

**Конец алгоритма**

1. **Проект программы с определением значения четвёртой, шестой и пятнадцатой степени числа**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из трёх частей: 1 блок ввода, 1 блок арифметических вычислений и 1 блок вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, полученных в процессе вычисления чисел и результата.

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



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <cstdlib>

using namespace std;

int main (int argc, char const \*argv[]) {

int a, x, y, z, temp\_x, temp\_y, temp\_z, temp1\_y, temp1\_z, temp2\_z, temp3\_z;

cout << "a=";

cin >> a;

temp\_x = a\*a; //2

x = temp\_x\*temp\_x; //4

temp\_y = a\*a; //2

temp1\_y = temp\_y\*temp\_y; //4

y = temp\_y\*temp1\_y; //6

temp\_z = a\*a; //2

temp1\_z = temp\_z\*a; //3

temp2\_z = temp1\_z\*temp1\_z; //6

temp3\_z = temp2\_z\*temp2\_z; //12

z = temp3\_z\*temp1\_z; //15

cout << "a^4=" << x <<endl;

cout << "a^6=" << y <<endl;

cout << "a^15=" << z <<endl;

return 0;

}

**на языке Pascal**

Program lab\_01\_1;

var a, x, y, z, temp\_x, temp\_y, temp\_z, temp1\_y, temp1\_z, temp2\_z, temp3\_z: integer;

Begin

Readln(a);

temp\_x:= a\*a; //2

x:= temp\_x\*temp\_x; //4

temp\_y:= a\*a; //2

temp1\_y:= temp\_y\*temp\_y; //4

y:= temp\_y\*temp1\_y; //6

temp\_z:= a\*a; //2

temp1\_z:= temp\_z\*a; //3

temp2\_z:= temp1\_z\*temp1\_z; //6

temp3\_z:= temp2\_z\*temp2\_z; //12

z:= temp3\_z\*temp1\_z; //15

Writeln(x);

Writeln(y);

Writeln(z);

readln;

end.

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

Для проверки работоспособности нужно просто запустить программу.

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

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.