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

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 3 «Итерационные 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 теоретических знаний конструкции выбора структурного программирования;
* Получение практических навыков разработки разветвляющихся алгоритмов;
* Построение логических выражений и использования условных операторов;

Задание:

Заданы два целых положительных числа X и Y. Найти значение , не используя операцию умножения.

**Вариант 30**

**Формальное описание задачи.**

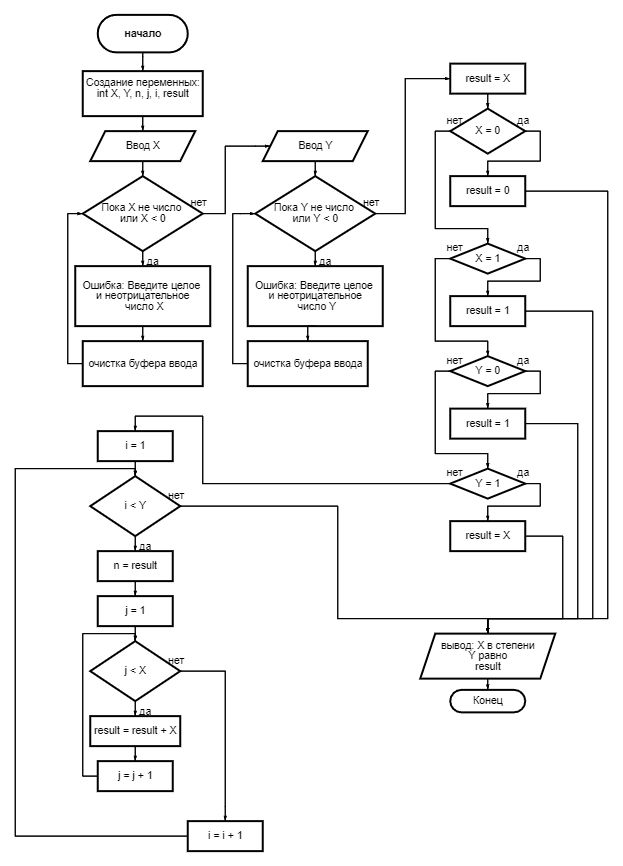
Пользователь вводит значение X и Y, а мы должны написать код, который возведет число X на число Y. Причем, мы не можем использовать операцию умножения, а должны сделать это при помощи циклов. Также мы должны сделать проверку на ввод значений.

**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввести исходные данные – число и степень, на которую нужно возвести.
2. При помощи условных операторов и циклов проверяем значения на ввод. Если ввод не правильный, то программа будет требовать значения, пока не получит правильные значения.
3. При помощи циклов производим операцию возведения в степень.
4. Вывести результаты на экран.

**Блок-схема алгоритма программы**



**Листинг программы**

\*/

Лабораторная работа 3

Вариант 30

*Заданы два целых положительных числа X и Y. Найти значение , не используя операцию умножения.*

\*/

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main() {

setlocale(LC\_ALL, "Rus");

int X, Y, n, j, i, result;

printf("Введите целые и неотрицательные числа X и Y, а программа выдаст вам результат возведения числа X в степень Y \n");

printf("Введите значение X: ");

while (scanf("%d", &X) != 1 || X < 0 ) {

printf("Ошибка: Введите целое и неотрицательное число X\n");

while (getchar() != '\n'); // очистка буфера ввода

}

printf("Введите значение Y: ");

while (scanf("%d", &Y) != 1 || Y < 0) {

printf("Ошибка: Введите целое и неотрицательное число Y\n");

while (getchar() != '\n'); // очистка буфера ввода

}

result = X;

if (X == 0){

result = 0;

} else if (X == 1){

result = 1;

} else if (Y == 0){

result = 1;

} else if (Y == 1){

result = X;

} else{

for(i = 1; i < Y; i++){

n = result;

for (j = 1; j < X; j ++){

result += n;

}

}

}

printf("%d в степени %d равно %d\n", X, Y, result);

return 0;

}

**Пояснения к программе.**

1. В этом коде мы определяем функцию main, которая является точкой входа в программу. Мы также включаем библиотеки stdio.h, stdlib.h и locale.h для работы с функциями ввода-вывода и числами.
2. Затем мы объявляем переменные X, Y, n, j, i и result как целочисленные. Мы используем переменные X и Y для хранения введенных пользователем чисел, а переменные n, j и i используются в циклах. Переменная result используется для хранения результата возведения X в степень Y.
3. Мы используем функцию setlocale для установки русского языка.
4. Мы используем функцию printf для вывода сообщения, объясняющего пользователю, что ему нужно ввести. Затем мы используем функцию scanf для ввода значений X и Y. Мы используем цикл while для проверки, что пользователь ввел корректные значения X и Y. Если пользователь вводит некорректное значение, мы выводим сообщение об ошибке и очищаем буфер ввода с помощью функции getchar.
5. Затем мы используем условные операторы if и else if для проверки значений X и Y. Если X равно 0, Y равно 0 или X равно 1 и Y равно 1, мы присваиваем переменной result соответствующее значение. В противном случае мы используем цикл for для возведения X в степень Y.
6. Наконец, мы используем функцию printf для вывода результата и функцию return для завершения программы.

**Вывод:**

Закрепил теоретические знания конструкций повторения структурного программирования, получил практические навыки разработки циклических алгоритмов, использовал операторы цикла.