Алгоритмические языки.

Лабораторная работа №1.

ИУ8-14; Заря В.Г.

**Цель работы**: написать две программы для решения математической и физической задач; попрактиковаться в задании переменных, расчетах и использовании логических операторов if/else, а также switch.

**Задача 1:** необходимо ввести значения некоторых переменных(ОДЗ от 1 до 4)-программа должна высчитать результат операций с этими числами, таких как возведение в степень и складывание.

**Код программы:**

#include <iostream>

#include <math.h>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int y, x, a, b, c; //Обьявляем переменные

cout << "\tВведите значения a,b,c,x. Разрешено вводить только числа от 1 до 4." << endl;

a:

cout << "a:";

cin >> a;

if (a<1 || a>4) {

cout << "Ошибка!" << endl; //Вводим значения переменных, если число не входит в одз, то программа скачет наверх, к вводу переменной.

goto a;

}

b:

cout << "\nb:";

cin >> b;

if (b<1 || b>4) {

cout << "Ошибка!" << endl;

goto b;

}

c:

cout << "\nc:";

cin >> c;

if (c<1 || c>4) {

cout << "Ошибка!" << endl;

goto c;

}

x:

cout << "\nx:";

cin >> x;

if (x<1 || x>4) {

cout << "Ошибка!" << endl;

goto x;

}

cout << "\nВведенное a:" << a; //показывает введенные значения

cout << "\nВведенное b:" << b;

cout << "\nВведенное c:" << c;

cout << "\nВведенное x:" << x << endl;

y = pow(a, x\*x - x + 5) + pow(b, x\*x - x + 4) - pow(c, x\*x + x + 2); // вычисляет некую операцию

cout << "\ny=" << y << endl;

system("pause");

return 0;}

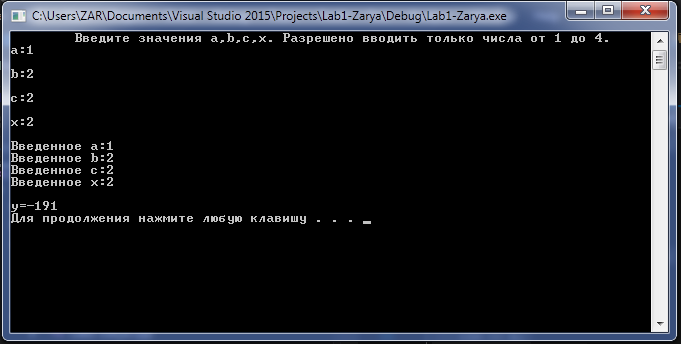
**Ручной расчёт:**

x=2,a=1,b=2,c=2;

Подставляем в выражение:

y= ;

y=1+-=1+= -191.

**Расчёт программой:**

**Выводы по задаче 1:** нужно включить в код библиотеку математических операций(#include <math.h>

**Задача 2:** из точки начала прямоугольной системы координат мы метаем тело. В программу задается начальная скорость снаряда, угол броска к горизонту, а также тип снаряда(определяет наличие/величину ускорения по оси х - нечто вроде сопротивления воздуха). Затем программа рассчитывает макс подъём снаряда по y и дальность полета по x. В программу нельзя вбить инвалидные значения скорости(с минусом), и углов.

**Код программы:**

#include <iostream>

#include <math.h>

#include<string>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

string ammoname[4] = { "Маленький металлический патрон", "Массивная тяжелая пуля", "Небольшое ядро","Массивное ядро" };

double v, angle, y, x, t;

int ammo;

const int g = 10;

int tx; //tx-время обращения скорости по x в ноль

/\*v-начальная скорость,angle-угол броска к горизонту,y-макс высота подьема,t-время достигания макс точки подьема,

\* x-дальность полета снаряда,ammotype-тип снаряда(определяет величину ускорения по оси x(типа сопротивление воздуха).\*/

angle\_choice:

cout << "Выберите угол,под которым запускают снаряд(от 0 до 90):";

cin >> angle;

if (angle<0 || angle>90) {

cout << "Ошибка!" << endl;

goto angle\_choice;

}

v\_choice:

cout << "Выберите скорость выстрела:";

cin >> v;

if (v<0) {

cout << "Ошибка!" << endl;

goto v\_choice;

}

cout << "Выберите тип снаряда:";

cout << "\n\t>>Введите '1',чтобы выбрать маленький металлический патрон";

cout << "\n\t>>Введите '2',чтобы выбрать массивную тяжелую пулю";

cout << "\n\t>>Введите '3',чтобы выбрать небольшое ядро";

cout << "\n\t>>Введите '4',чтобы выбрать массивное ядро";

ammo\_choice:

cout << "\nВыберите тип снаряда:";

cin >> ammo;

switch (ammo) {

case 1: {

y = v\*v\*sin(angle)\*sin(angle) / 2 / g;

t = v\*sin(angle) / g;

x = 2 \* v\*cos(angle)\*t;

goto final; }

break;

case 2: {

int b = 1; //ускорение по оси х

tx = v\*cos(angle) / b; //время,которое нужно по оси x для остановки(невзирая на ось y)

t = v\*sin(angle) / g;

if (2 \* t<tx) { //если 2\*t меньше чем tx,то к моменту касания снарядом земли скорость по x не успеет обратиться в 0 или в отрицательную.

x = 2 \* v \* cos(angle) \* t - b \* 2 \* pow(t, 2); //x в таком случае зависит от времени t

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;

goto final; //y независимый

}

else { //Здесь уже не важно,будет ли программа считать,что скорость по x может быть отрицательной.Ведь даже в таком случае никакого влияние на движение по оси y не оказывается.

x = v \* v \* cos(angle) \* cos(angle) / 2 / b; //x не зависит от t

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;//y независимый;даже если скорость по x обратится в 0 или в отрицательную,то x достигает макс значения не оказывая влияния на движение по оси y.

goto final;

//case 3 и case 4 содержат измененное значение b-ускорения по оси х,а все прочие формулы наследуются.

}

}

break;

case 3: {

int b = 4;

tx = v\*cos(angle) / b;

t = v\*sin(angle) / g;

if (2 \* t<tx) {

x = 2 \* v \* cos(angle) \* t - b \* 2 \* pow(t, 2);

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;

goto final;

}

else {

x = v \* v \* cos(angle) \* cos(angle) / 2 / b;

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;

goto final;

}

}

break;

case 4: {

int b = 11;

int tx;

tx = v\*cos(angle) / b;

t = v\*sin(angle) / g;

if (2 \* t<tx) {

x = 2 \* v \* cos(angle) \* t - b \* 2 \* pow(t, 2);

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;

goto final;

}

else {

x = v \* v \* cos(angle) \* cos(angle) / 2 / b;

y = v \* v \* sin(angle) \* sin(angle) / 2 / g;

goto final;

}

}

break;

default:

cout << "Ошибка!" << endl;

goto ammo\_choice;

}

final:

cout << "\nЗаданные значения:" << endl;

cout << "Начальная скорость:" <<v<<endl;

cout << "Угол броска к горизонту:" << angle << endl;

cout << "Тип снаряда:";

switch (ammo) {

case 1: {

cout << ammoname[0];

goto finals;

}

break;

case 2: {

cout << ammoname[1];

goto finals;

}

break;

case 3: {

cout << ammoname[2];

goto finals;

}

break;

case 4: {

cout << ammoname[3];

goto finals;

}

break;

}

finals:

cout << "\nМаксимальный подьём:" << y << endl;

cout << "Дальность полета:" << x << endl;

system("pause");

return 0;

}

**Ручной расчет:**

Возьмем начальную скорость равной 10, угол броска условимся считать равным 30. Решим задачу для двух случаев:

1) Ammotype = 1. Тогда имеем классическую задачу, где движение по оси x равномерное, а движение по y - равноускоренное.

2)Ammotype = 3. Тогда имеем также ускорение по оси x, что уменьшит дальность полета.