Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Институт информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно-ориентированное программирование»

**ОТЧЕТ**

По лабораторной работе № 1

Тема: «Повторение»

Выполнила: студентка группы

ПИ(б)-3-21 Ашымова А.А

Проверил: Мусабаев Э.Б

Бишкек – 2024

**Задание 1**

Возведение числа **n** в степень **p** – это умножение числа **n** на себя **p** раз.

Напишите функцию (не библиотечную) с именем **power** (), которая в качестве аргументов принимает значение типа **double** для **n** и значение типа **int** для **p** и возвращает значение типа **double**.

Для аргумента, соответствующего степени числа, задать значение по умолчанию, равное **2**, чтобы при отсутствии показателя степени при вызове функции число **n** возводилось в квадрат.

Сделайте проверку, будет ли пользователь вводить степень числа. Если будет, то эта степень – число, большее 2. Напишите функцию **main**(), которая запрашивает у пользователя ввод аргументов для функции **power** () и отображает на экране результаты её работы. Напишите код первого задания с *объявлением* (прототипом) функции power ().

Постановка задачи:

Входные переменные:

x - основание.

y - степень числа.

Выходные параметры:

z - результат.

// Задание 1

#include<iostream>

using namespace std;

// прописываем прототип функции

double Power(double, int );

int main() {

setlocale(LC\_ALL,"rus");

// объявляем переменные

double x, z;

int y=2;

// прописываем бесконечный цикл для удобства

while (true) {

// вводим значения

sno:cout << "Введите значение x: ";

cin >> x;

//если х=1 бесконечный цикл завершает работу программы

if (x == 1) {

break;

}

cout << "Введите степень y: ";

cin >> y;

// используем гоу ту если введено неверное значение

// при использовании оператора мы переходим к метке

if (y <= 2) {

cout << "Введите число больше 2!!!" << endl;

goto sno;

}

// вызов функции и принятие параметров

z = Power(x, y);

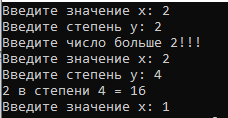
// возвращает значение которое сразу же распечатываем

cout << x << " в степени " << y << " = " << z << endl;

}

system("pause");

return 0;

}

// наш блок кода которые вызывается в мэин.

// определяем его

double Power(double a, int b) {

double c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

**Задание 2**

Взяв в качестве основы функцию **power** () из первого задания, работающую только со значением типа **double**. Создайте перегруженные функции с этим же именем, принимающими в качестве аргумента значения типа **char**, short **int**, **long int, float**. Напишите программу, вызывающую функцию **power** () со всеми возможными типами аргументов. Напишите код второго задания без *объявления* функции power ().

// Задание 1

#include<iostream>

using namespace std;

// если функция пишется до мейн, то ее объявлять не надо

double Power(double a, int b) {

double c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

char Power(char a, int b) {

char c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

short int Power(short int a, int b) {

short int c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

long int Power(long int a, int b) {

long int c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

float Power(float a, int b) {

float c = 1;

for (int i = 1; i <= b; i++) {

c \*= a;

}

return c;

}

int main() {

setlocale(LC\_ALL, "rus");

double x, z;

char x1, z1;

short int x2, z2;

long int x3, z3;

float x4, z4;

int y = 2;

cout << "Введите double x от -1.7e308 до 1.7e308: ";

cin >> x;

cout << "Введите char x от -128 до 127: ";

cin >> x1;

cout << "Введите short int x от -32,768 до 32,767.: ";

cin >> x2;

cout << "Введите long int x от -2,147,483,648 до 2,147,483,647: ";

cin >> x3;

cout << "Введите float x от -3.4e38 до 3.4e38: ";

cin >> x4;

sno:cout << "Введите степень y: ";

cin >> y;

if (y <= 2) {

cout << "Введите число больше 2!!!" << endl;

goto sno;

}

// вызов функций и вывод результата

// для double

z = Power(x, y);

cout << x << " в степени " << y << " = " << z << endl;

// для char

z1 = Power(x1, y);

cout << x1 << " в степени " << y << " = " << z1 << endl;

// для char

z2 = Power(x2, y);

cout << x2 << " в степени " << y << " = " << z2 << endl;

// для char

z3 = Power(x3, y);

cout << x3 << " в степени " << y << " = " << z3 << endl;

// для char

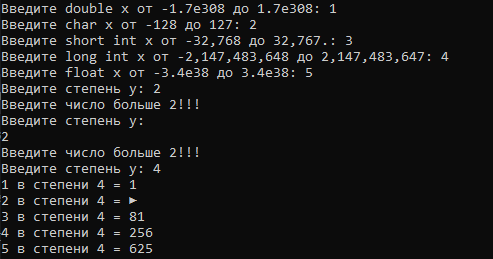
z4 = Power(x4, y);

cout << x4 << " в степени " << y << " = " << z4 << endl;

system("pause");

return 0;

}



**Задание 3**

Информация по работникам N заводов задается структурой с полями: фамилия, возраст, специальность, средний оклад. Ввести информацию по заводам, посчитать количество слесарей и токарей. Вывести эти значения на консоль.

// Задание 1

#include<iostream>

using namespace std;

// прописываем структуру

struct Rabotnic {

string FIO;// фио

int age;//возраст

string spec;//специальность

double ocl;//средний оклад

};

int main() {

setlocale(LC\_ALL, "rus");

// создаем переменные типа работник

const int n = 3;

int c = 0, v = 0;

Rabotnic a[n];

a[0].FIO = "Токарев Иван";

a[0].age = 20;

a[0].spec = "Токарь";

a[0].ocl = 10000;

a[1] = { "Токаренко Наталья",

30,

"Токарь",

20000 };

a[2] = { "Слесарева Наташа",

40,

"Слесарь",

30000 };

for (int i = 0; i < n; i++) {

if (a[i].spec == "Токарь") {

c++;

}

else if (a[i].spec == "Слесарь") {

v++;

}

}

cout << "Количество слесарей: " << c << endl;

cout << "Количество токарей: " << v << endl;

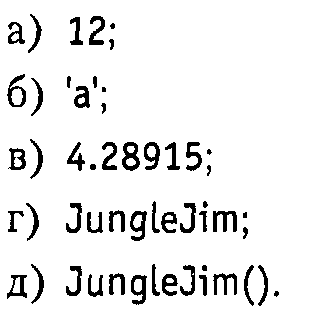
system("pause");

return 0;

}



**ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ №1**

1. Всего 5 простых типов: int, char, bool, float, double.
2. a – значение которое целоч-ой присваиваемая переменной.

б – значение для переменной типа char.

в - значение для переменной типа double.

г - не знаю

д –метода

1. Функция – именованная последовательность описаний и операторов, выполняющих какие-либо законченные действия.  
   Функция – блок программного кода, не входящий в основную программу и выполняющий определенную задачу.
2. Заголовок функции (<тип функции> , <имя функции>, <(список параметров)> {

Тело функции (код)

}

1. Есть 2 способа: 1 объявляется прототип (чтобы компилятор знал, что такая функция имеется) и пишется перед меин, а саму функцию определяем уже после мейна.

2 перед мейн сразу прописывается функция и определяется и надобность в прототипе отпадает

1. Разница в том что в 1 копия, а вторая адрес.
2. По ссылке, указателю и значению и по умолчанию.
3. Если мы не указываем аргумент, то используется аргумент по умолчанию.
4. Механизм, благодаря которому функции с разным количеством и типом параметров могут иметь одно имя.
5. В структуре каждое поле может иметь разный тип данных, в то время как все элементы массива имеют один и тот же тип данных