Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им.И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Объектно-ориентированное программирование»

Лабораторная работа №1

Тема: «Повторение»

Выполнил: студент группы

ПИ(б)-2-19 Улан уулу Нурдин

Проверила: Каткова С. Н.

Бишкек – 2021

**Задание 1**

1. **Постановка задачи:**

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

1. 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**.
2. Для аргумента, соответствующего степени числа, задать **значение по умолчанию**, равное **2**, чтобы при отсутствии показателя степени при вызове функции число **n** возводилось в квадрат.
3. Сделайте проверку, будет ли пользователь вводить степень числа. Если будет, то эта степень – число, большее 2.
4. Напишите функцию **main**(), которая запрашивает у пользователя ввод аргументов для функции **power** () и отображает на экране результаты её работы.
5. Напишите код первого задания с *объявлением* (прототипом) функции power ().
6. Начертите блок-схему работы созданного Вами программного кода.
7. **Описание исходных данных:**

**Переменные**

* Локальная переменная **value** типа **double** – хранит в себе число;
* Локальная переменная **valuePower** типа **int** – хранит в себе степень числа;

**Функции**

* double power(double n, int p = 2) – функция возвращающая степень числа, которые введет пользователь;

1. **Блок-схемы:**

int main()



double power(double n, int p = 2)



1. **Код программы:**

#include <iostream>

#include <ctime>

#include <fstream>

#include <string>

#include "Windows.h"

#include <time.h>

#include <iomanip>

#define ll long long

#define str string

using namespace std;

double power(double n, int p = 2);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double value;

int valuePower;

cout << "Введите число: ";

cin >> value;

cout << "Введите степень в которую возвести: ";

cin >> valuePower;

if (valuePower < 2) {

cout << "Вы ввели степень < 2, из за этого степень будет равна 2!!!" << endl;

cout << "Число " << value << " в степени 2 равна " << power(value) << endl;

} else {

cout << "Число " << value << " в степени " << valuePower << " равна " << power(value, valuePower) << endl;

}

return 0;

}

double power(double n, int p) {

double temp = 1;

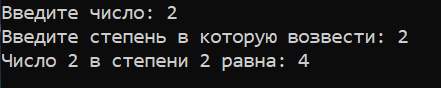
for (int i = 0; i < p; i++)

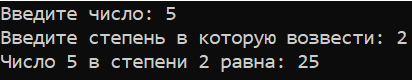
temp \*= n;

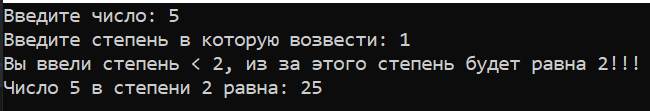
return temp;

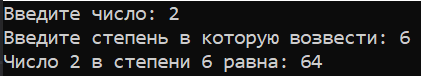
}

1. **Результаты работы программы:**









**Задание 2**

1. **Постановка задачи:**

Взяв в качестве основы функцию **power** () из первого задания, работающую только со переменными типа double, cоздайте перегруженные функции с этим же именем, работающие с переменными типа char, short int, long int, float.

1. Напишите программу, вызывающую функцию **power** () со всеми возможными типами аргументов.
2. Напишите код второго задания **без *объявления* функции power ().**
3. Начертите блок-схему работы созданного Вами программного кода.
4. **Описание исходных данных:**

**Переменные**

* Локальная переменная **ShortValue** типа **short int** – хранит в себе число;
* Локальная переменная **LongValue** типа **long int** – хранит в себе число;
* Локальная переменная **DoubleValue** типа **double** – хранит в себе число;
* Локальная переменная **FloatValue** типа **float** – хранит в себе число;
* Локальная переменная **Power** типа **int** – хранит в себе степень числа;
* Локальная переменная **x** типа **int** – хранит в себе индекс меню;
* Локальная переменная **s** типа **string** – хранит в себе название используемой переменной;
* Локальная переменная **f** типа **bool** – для основного цикла меню;

**Функции**

* short int power(short int n, int p = 2) – функция возвращающая степень целочисленного числа, которые введет пользователь;
* long int power(long int n, int p = 2) – функция возвращающая степень целочисленного числа, которые введет пользователь;
* double power(double n, int p = 2) – функция возвращающая степень вещественного числа, которые введет пользователь;
* float power(float n, int p = 2) – функция возвращающая степень вещественного числа, которые введет пользователь;
* void menu() – функция для вывода меню;
* void output(T& value, int& valuePower, str s) – функция для вызова функции power(), а так же ввода и вывода числа и его степени;

1. **Блок-схемы:**

int main()



double power(double n, int p = 2)



void output(T& value, int& valuePower, str s)



1. **Код программы:**

#include <iostream>

#include <ctime>

#include <fstream>

#include <string>

#include "Windows.h"

#include <time.h>

#include <iomanip>

#define ll long long

#define str string

using namespace std;

double power(double n, int p = 2) {

double temp = 1;

for (int i = 0; i < p; i++)

temp \*= n;

return temp;

}

short int power(short int n, int p = 2) {

short int temp = 1;

for (int i = 0; i < p; i++)

temp \*= n;

return temp;

}

long int power(long int n, int p = 2) {

long int temp = 1;

for (int i = 0; i < p; i++)

temp \*= n;

return temp;

}

float power(float n, int p = 2) {

float temp = 1;

for (int i = 0; i < p; i++)

temp \*= n;

return temp;

}

template <class T>

void output(T& value, int& valuePower, str s) {

cout << "Введите число типа " << s << ": ";

cin >> value;

cout << "Введите степень в которую возвести: ";

cin >> valuePower;

system("cls");

if (valuePower < 2) {

cout << "Вы ввели степень < 2, из за этого степень будет равна 2!!!" << endl;

cout << "Число " << value << " типа " << s << " в степени 2 равна: " << power(value) << "\n\n\n";

} else

cout << "Число " << value << " типа " << s << " в степени " << valuePower << " равна: " << power(value, valuePower) << "\n\n\n";

}

void menu() {

cout << "-----------------------------------------" << endl;

cout << "| Главное Меню: |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 1. Short |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 2. Long |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 3. Double |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 4. Float |" << endl;

cout << "-----------------------------------------" << endl;

cout << "| 5. Выход. |" << endl;

cout << "-----------------------------------------" << endl;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int Power, x;

short int ShortValue;

long int LongValue;

double DoubleValue;

float FloatValue;

str s; bool f = true;

while (f) {

menu();

cout << "\nВыберите тип данных: ";

cin >> x;

system("cls");

switch (x) {

case 1: s = "short"; output(ShortValue, Power, s); break;

case 2: s = "long"; output(LongValue, Power, s); break;

case 3: s = "double"; output(DoubleValue, Power, s); break;

case 4: s = "float"; output(FloatValue, Power, s); break;

case 5: f = false; break;

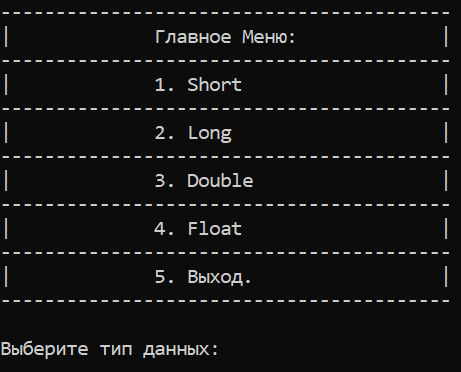
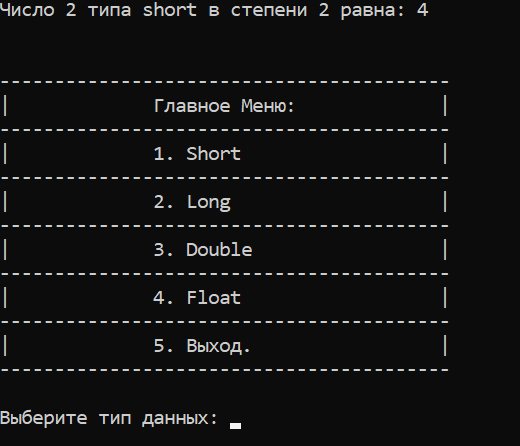
}

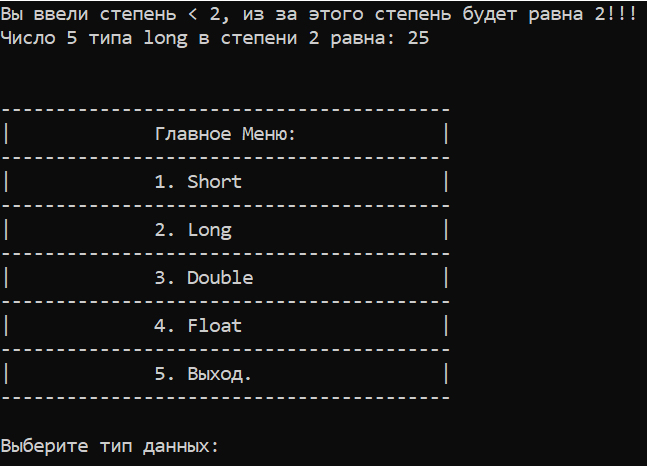
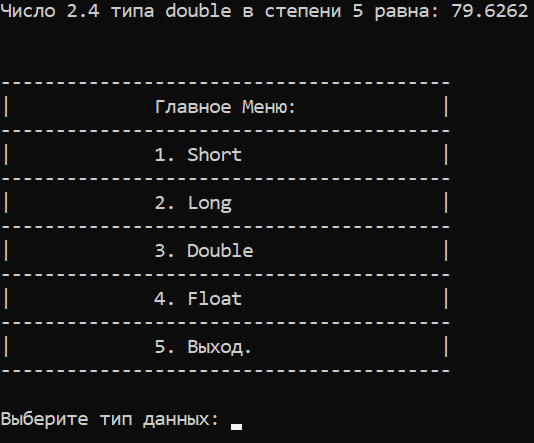
}

return 0;

}

1. **Результаты работы программы:**

**Задание 3**

1. **Постановка задачи:**

Информация по работникам N заводов задается структурой с полями: фамилия, возраст, специальность, размер зарплаты. Ввести информацию по всем заводам, посчитать количество слесарей и токарей. Вывести эти значения на консоль. Начертите блок-схему работы созданного Вами программного кода.

1. **Описание исходных данных:**

**Переменные**

* Локальная переменная **n** типа **int** – хранит в себе количество работников;
* Локальная переменная **countLocksmith** типа **int** – хранит в себе количество слесарей;
* Локальная переменная **countTurner** типа **int** – хранит в себе количество токарей;
* Структура Plant описывает работников завода, и состоит из следующих полей:

**name** – переменная тип **string**, хранящая в себе ФИО работника;

**age** – переменная тип **int**, хранящая возраст работника;

**specialty** – переменная тип **string**, хранящая в себе специальность работника;

**salary** – переменная тип **int**, хранящая заработную плату работника;

1. **Блок-схемы:**

int main()



1. **Код программы:**

#include <iostream>

#include <ctime>

#include <fstream>

#include <string>

#include "Windows.h"

#include <time.h>

#include <iomanip>

#define ll long long

#define str string

using namespace std;

struct Plant {

str name;

int age = 0;

str specialty;

int salary = 0;

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

fstream sin;

int n, countLocksmith = 0, countTurner = 0;

sin.open("0\\Plant.txt");

sin >> n;

Plant\* arr = new Plant[n];

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

sin.ignore(); getline(sin, arr[i].name);

sin >> arr[i].age;

sin.ignore(); getline(sin, arr[i].specialty);

sin >> arr[i].salary;

}

sin.close();

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

if (arr[i].specialty == "Locksmith")

countLocksmith++;

else if (arr[i].specialty == "Turner")

countTurner++;

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

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

cout << "| " << i + 1 << " | " << setw(19) << arr[i].name << " | " << arr[i].age << " | " << setw(10) << arr[i].specialty << " | " << arr[i].salary << " |" << endl;

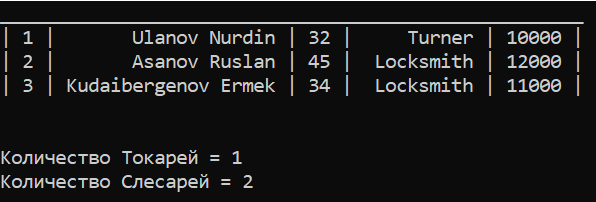
cout << "\n\nКоличество Токарей = " << countTurner << "\nКоличество Слесарей = " << countLocksmith << endl;

delete[]arr;

return 0;

}

1. **Результаты работы программы:**

****

**Ответы на вопросы:**

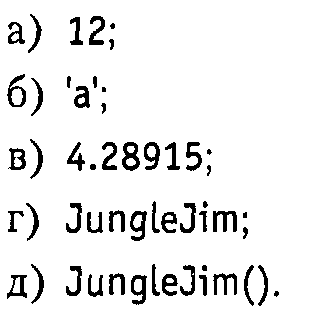
1. **Какие стандартные типы данных используются в С++? Сколько под них резервируется памяти?**

Целочисленные (int) – 4 байта;

Вещественные (double) – 8 байтов;  
Символьные (char) – 1 байт;

Логические (bool) – 1 байт.

1. **К каким элементам программы относятся следующие:**



* 1. – int;

(б) - char;

(в) – double;

(г) – переменная;

(д) – функция.

1. **Что такое функция? Какова роль функций в языке С++?**

Функция — фрагмент программного кода, к которому можно обратиться из другого места программы. С именем функции неразрывно связан адрес первой инструкции, входящей в функцию, которой передаётся управление при обращении к функции. После выполнения функции управление возвращается обратно в адрес возврата — точку программы, где данная функция была вызвана.

1. **Напишите синтаксис функции.**

Тип\_функции Название\_Функции (Передаваемые\_данные); - прототип;

Тип\_функции Название\_Функции (Передаваемые\_данные) {

Тело\_функции;

} – без прототипа.

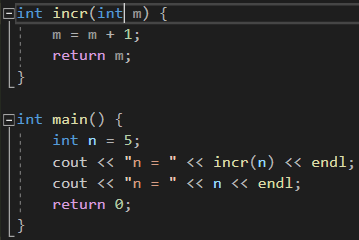
1. **Опишите способы использования функций в программах с объявлением функций (с прототипом) и без объявления функций (без прототипа).**

Функция с прототипом может быть вызвана до ее реализации, то есть описания самого кода функции.

1. **Объясните механизмы передачи аргументов по значению и по ссылке в функцию.**

Передача в функцию по значению создает копию переменной, которую передают в функцию и не может быть изменена в функции, а по ссылке отправляется сама переменная, и может быть изменена в функции.

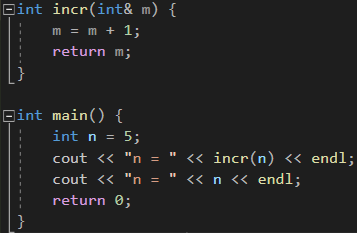
**Объясните результаты работы программ.**

* 1. ****

Вывод будет: n = 6

n = 5

Потому что в функцию передается значение n = 5, в функции создается ее копия и увеличивается на 1, и возвращается в main и выводиться 6, но поскольку в функции увеличилась копия, а не оригинал при выводе n будет равен так же 5.

* 1. 

Вывод будет: n = 6

n = 6

Потому что в функцию передается значение по ссылке n = 5, то есть сама переменная, а не ее копия и увеличивается на 1, и возвращается в main и выводиться 6 два раза, так как и в функции увеличилась m и вернулась в main так и n, потому что с ней происходит то же что и с m.

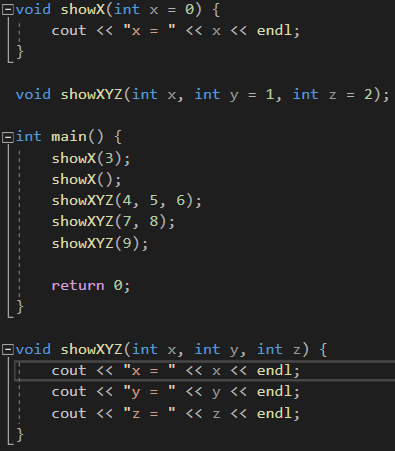
1. **Назовите разновидности аргументов, которые могут быть переданы параметрам функции?**

Аргументы функции могут быть переданы по ссылке, указателю и по значению.

1. **Что такое аргументы по умолчанию?**

В программировании, аргумент по умолчанию является аргументом функции, который при вызове необязательно указывать. В большинстве языков программирования, функции могут принимать один или несколько аргументов. Как правило, все аргументы должны быть указаны в полном объеме.  Позже некоторые языки, например, C++ позволяли программисту в объявлении функции указывать значения некоторых аргументов. Таким образом функцию можно было вызывать передавая лишь обязательные аргументы.

**Объясните результат работы программы:**

****

Будет выведено: x = 3

x = 0

x = 4, y = 5, z = 6

x = 7, y = 8, z = 9

x = 9, y = 1, z = 2

Вначале будет выведено 3 потому что мы в main в функцию showX() передается 3, далее выводиться 0, потому что значение по аргументу, то есть если мы ничего не передадим в функцию showX() то x будет равен 0, далее будет выведено x = 4, y = 5, z = 6, потому что мы передаем в функцию showXYZ() 4,5 и 6, далее так же как и с функцией showX(), аргументы по умолчанию ставятся туда куда не хватает значений.

1. **Что такое перегрузка функций?**

Перегрузка функции это одно и тоже название у двух функций, но принимающие разные аргументы.

1. **Чем отличается структура (struct) от массива?**

Структура может хранить в себе разные типы данных, а массив только один тип данных.