**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И.Раззакова**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра: **Программное обеспечение компьютерных систем**

Курс «Объектно-ориентированное программирование»

**ОТЧЕТ**

**Лабораторная работа №1**

Выполнил: студент группы ПИ-3-21

Закиров Мустафа

Проверил: Мусабаев Э.Б.

**Бишкек 2024**

# **Задание №1**

**Постановка задачи:**

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.

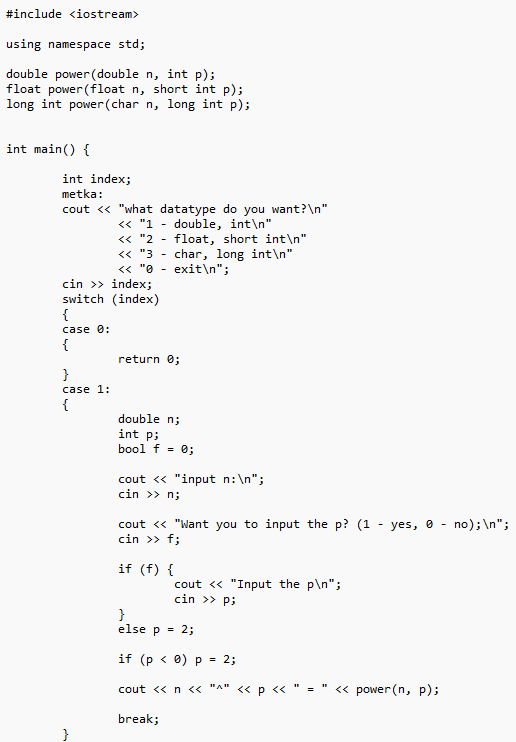
Сделайте проверку, будет ли пользователь вводить степень числа. Если будет, то эта степень – число, большее 2. Напишите функцию **main**(), которая запрашивает у пользователя ввод аргументов для функции **power** () и отображает на экране результаты её работы. Напишите код первого задания с *объявлением*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

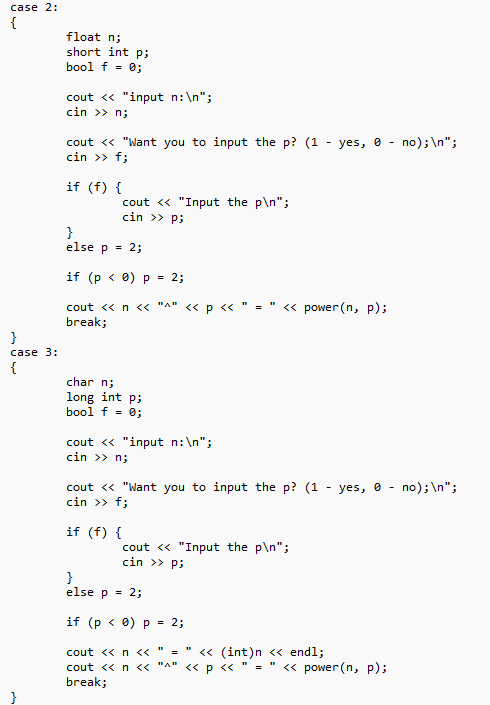
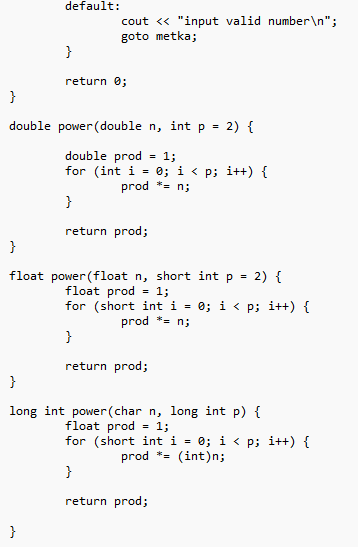
Взяв в качестве основы функцию **power** () из первого задания, работающую только со значением типа **double**. Создайте перегруженные функции с этим же именем, принимающими в качестве аргумента значения типа **char**, short **int**, **long int, float**. Напишите программу, вызывающую функцию **power**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

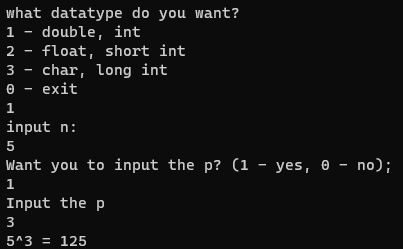
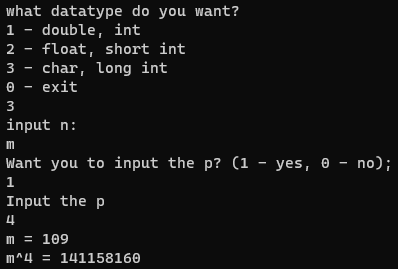
**Входные данные:**

В качестве входных данных нам даны n, p.

**Исходный код программы на C++:**

****



**Тесты:**

# **Задание №3**

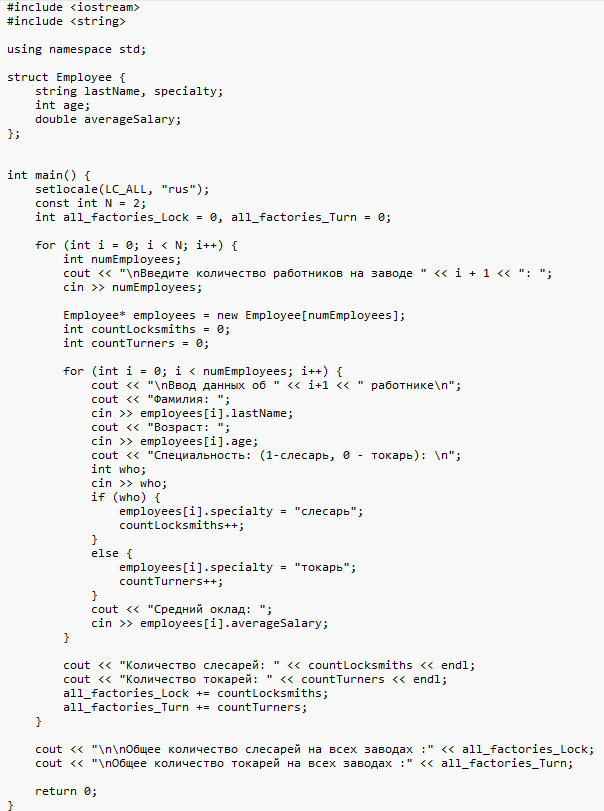
**Постановка задачи:**

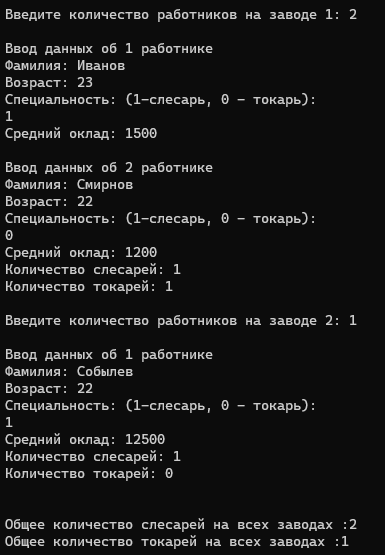
Информация по работникам N заводов задается структурой с полями: фамилия, возраст, специальность, средний оклад. Ввести информацию по заводам, посчитать количество слесарей и токарей. Вывести эти значения на консоль.

**Исходные данные:**

N – количество заводов.

**Исходный код программы на C++:**

**Исходный код программы на C++:**

**Тест:**

# **Ответы на вопросы:**

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

• Bool – предназначен для хранения логических значений и требует 1 байта памяти;

• Char – служит для хранения символьных значений и занимает 1 байт памяти;

• Int – используется для представления целочисленных значений и резервирует 4 байта памяти;

• Short – применяется для целочисленных значений, требует 2 байта памяти;

• Long – служит для хранения целочисленных значений и выделяет 4 байта памяти;

• Float – применяется для числовых значений с плавающей точкой, использует 4 байта памяти;

• Double – предназначен для числовых значений с плавающей точкой двойной точности и требует 8 байт памяти.

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

а) 12;  
б) ‘a’;  
в) 4.28915;  
г) JungleJim;  
д) JungleJim().

а) 12 - это числовая константа с типом данных int.

б) 'a' - символьная константа с типом данных char.

в) 4.28915 - числовая константа с плавающей точкой и типом данных float.

г) "JungleJim" - строковый литерал, который может быть использован как значение переменной типа string или как идентификатор класса, функции, объекта и т. д.

д) JungleJim() - это объявление функции или конструктора с именем JungleJim и может также представлять собой вызов функции.

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

Функция представляет собой участок кода, предназначенный для выполнения конкретных задач в рамках своего определения. Она представляет собой отдельный блок алгоритма, шаг за шагом преобразующий входные данные в выходные, и может быть повторно использована в различных частях программы.

**4. Напишите синтаксис функции.**

#include <iostream>  
using namespace std;   
void printHello();  
int main() {  
 printHello   
 return 0;  
}  
void printHello(){ cout << "Hello, World!" << endl;   
}

**5. Опишите способы использования функций в программах с объявлением функций и без объявления функций.**

Предварительное объявление функции дает компилятору понять, что где-то в проекте есть функция с идентичным названием и входными параметрами.

void printHello();  
  
int main() {  
 printHello();  
 return 0;  
}  
  
void printHello() {  
 cout << "Hello, World!" << endl;  
}

Без объявления функции компилятор не сможет понять, что существует функция, пока не дойдет до нее, поэтому вызов функции, которая определена после функции main приведет к ошибке.

int main() {  
 printHello();  
 return 0;  
}  
  
void printHello() {  
 cout << "Hello, World!" << endl;  
}

Для корректной работы без объявления функции требуется определить функцию до ее вызова.

void printHello() {  
 cout << "Hello, World!" << endl;  
}  
  
int main() {  
 printHello();  
 return 0;  
}

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

Объясните результаты работы программ:

1. #include <iostream>

using namespace std;

int incr(int m){

m = m + 1;

return m;

}

int main() {

int n = 5;

cout << "n = " << incr(n) << endl;

cout << "n = " << n << endl;

return 0;

}

В начале объявляется функция incr с входным параметром m типа int, она возвращает значение m+1. Т.к. функция принимает входные данные по значению, то это не отразится на реальной переменной, в этом случае, m не изменится на +1.

Программа начинается с инициализации переменной n типа int равной 5. Далее идет операция вывода с вызовом функции incr(n), которая выведет число 6. Далее идет вывод числа n без использования функции, поэтому будет выведено число 5.

2.

#include <iostream>  
using namespace std;  
int incr(int &m){  
 m = m + 1;  
 return m;  
}  
int main() {  
 int n = 5;  
 cout << "n = " << incr(n) << endl;  
 cout << "n = " << n << endl;  
 return 0;  
}

В начале объявляется функция incr с входным параметром m типа int, она возвращает значение m+1. Т.к. функция принимает аргументы по ссылке, то это напрямую перезапишет значение переменной n из функции main.

Программа начинается с инициализации переменной n типа int равной 5. Далее идет операция вывода с вызовом функции incr(n), которая выведет число 6. Далее идет вывод числа n без использования функции, т.к. в прошлом вызове функции incr(n) значение n изменилось на 6, будет выведено 6.

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

Передача по значению. В функцию передается копия объекта, с которой она работает. После отработки функции значение оригинальной переменной остается неизменной.

void inc(int x) {  
 x = x + 1;  
}  
int main() {  
 int x = 5;  
 inc(x);  
 cout << x;  
 return 0;  
}

Передача по указателю. Вызов функции вместо копии переменной создаст переменную ссылочного типа x, в нее будет скопирован адрес ячейки x операцией взятия адреса. Далее, в функции, с помощью операции косвенной адресации в ячейку x запишется новый результат, который поменяет переменную, использующуюся в main.

void inc(int \*x) {  
 \*x = \*x + 1;  
}  
int main() {  
 int x = 5;  
 inc(&x);  
 cout << x;  
 return 0;  
}

Передача по ссылке. В случае вызова по ссылке оператор вызова дает вызываемой функции возможность прямого доступа к передаваемым данным, а также возможность изменения этих данных. Вызов по ссылке хорош в смысле производительности, потому что он исключает накладные расходы на копирование больших объемов данных.

void inc(int &x) {  
 x = x + 1;  
}  
int main() {  
 int x = 5;  
 inc(x);  
 cout << x;  
 return 0;  
}

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

Аргументы по умолчанию – это способ способность работы функции в случае, если в функцию не были переданы аргументы при вызове функции.

void printX(int x = 10) {  
 cout << x << endl;  
}  
int main() {  
 int x = 5;  
 printX(x);  
 printX();  
 return 0;  
}

Объясните результат работы программы:

#include <iostream>  
using namespace std;  
// Аргумент функции имеет значение по умолчанию  
void showX(int x = 0){  
 cout << "x = " << x << endl;  
}  
void showXYZ(int x, int y = 1, int z = 2);  
// Два аргумента функции в прототипе имеют значения  
// по умолчанию  
// сама функция описана в конце программы  
int main() {  
 showX(3);  
 showX();  
 showXYZ(4, 5, 6);  
 showXYZ(7, 8);  
 showXYZ(9);  
 return 0;  
}  
// При описании функции значения по умолчанию не указываются  
void showXYZ(int x, int y, int z){  
 cout << "x = " << x << " ";  
 cout << "y = " << y << " ";  
 cout << "z = " << z << endl;  
}

В начале инициализируется функция showX, которая принимает переменную x типа int и выводит ее. Она имеет значение 0 по умолчанию.

После идет определение функции showXYZ с 3 входными переменными: x типа int, y типа int со значением по умолчанию 1, z типа int со значением по умолчанию 2.

В конце после функции main определяется функция showXYZ, в которой идет вывод переменных x, y, z.

В функции main идет последовательный вызов функций:  
вызов функции showX(3), которая выведет 3;  
вызов функции showX() без входного аргумента, которая выведет 0 в консоль;  
вызов функции showXYZ(4, 5, 6), которая выведет “x = 4, y = 5, y = 6”$  
вызов функции showXYZ(7, 8), которая выведет “x = 7, y = 8, z = 2”, т.к. для z не передавалось значение и оно было установлено по умолчанию;  
вызов функции showXYZ(9), которая выведет «x = 9, y = 1, z = 2», т.к. для y, z не передавались значения и они были установлены по умолчанию.

**9. Что такое перегрузка функций?**

Перегрузка функции – возможность создания и использования функций с одним названием, но разными параметрами. При вызове функции компилятор автоматически выбирает правильную для использования функцию на основе количества входных параметров и их типах.

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

Массив (Array) – это упорядоченный набор последовательно расположенных однотипных элементов. В массиве все элементы имеют один и тот же тип данных и обращение к ним реализуется при помощи индексов.

Структура (Struct) – пользовательский тип данных, задаваемый программистом на свое усмотрение. В отличие от массива, структура может содержать различные типы данных, которые называются полями.