**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

Копжашаров Азамат

Проверил: Мусабаев Э.Б.

**Бишкек 2021**

# **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 ().

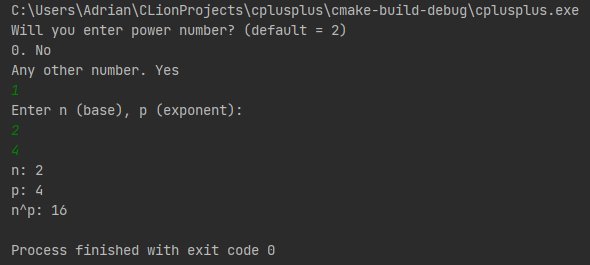
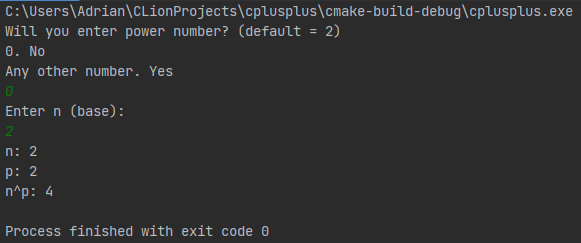
**Исходные данные:**

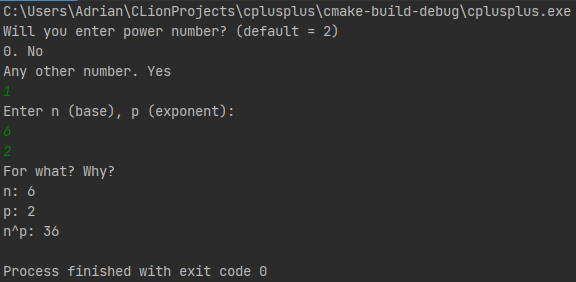
В качестве исходных данных нам даны n, p.

**Исходный код программы на C++:**

// 1.1  
#include <iostream>  
using namespace std;  
  
double power(double n, int p = 2);  
  
int main() {  
 int check;  
 int n, p, result;  
 cout << "Will you enter power number? (default = 2)" << endl  
 << "0. No \nAny other number. Yes" << endl;  
 cin >> check;  
 if(check){  
 cout << "Enter n (base), p (exponent): " << endl;  
 cin >> n >> p;  
 result = power(n, p);  
 if(p == 2) cout << "For what? Why?" << endl;  
 }else{  
 cout << "Enter n (base): " << endl;  
 cin >> n;  
 p = 2;  
 result = power(n);  
 }  
 cout << "n: " << n << endl  
 << "p: " << p << endl  
 << "n^p: " << result << endl;  
 return 0;  
}  
  
double power(double n, int p) {  
 double ans = n;  
 while(--p) ans \*= n;  
 return ans;  
}

**Тесты:**





# **Задание №2**

**Постановка задачи:**

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 ().

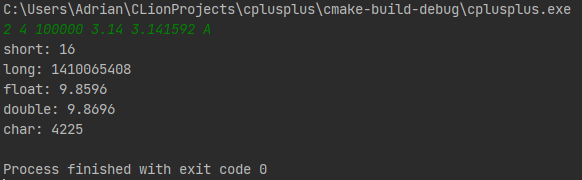
**Исходные данные:**

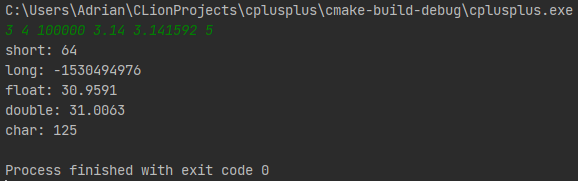
В качестве исходных данных нам ничего не дается, но исходя из условий задачи, требуется использовать аналогичные исходные данные, что и в задаче №1: p, n, только уже для каждого перечисленного типа данных.

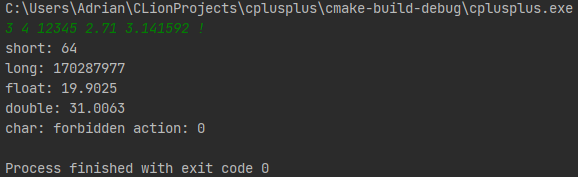
**Исходный код программы на C++:**

// 1.2  
#include <iostream>  
using namespace std;  
  
double power(double n, int p = 2) {  
 double ans = n;  
 while (--p) ans \*= n;  
 return ans;  
}  
  
short power(short n, int p = 2) {  
 short ans = n;  
 while (--p) ans \*= n;  
 return ans;  
}  
  
long power(long n, int p = 2) {  
 long ans = n;  
 while (--p) ans \*= n;  
 return ans;  
}  
  
float power(float n, int p = 2) {  
 float ans = n;  
 while (--p) ans \*= n;  
 return ans;  
}  
  
int power(char n, int p = 2) {  
 if((int)n >= 48 && (int)n <= 57){  
 int ans = (int)(n-48), exp = (int)(n-48);  
 while(--p) ans \*= exp;  
 return ans;  
 }else if(((int)n >= 65 && (int)n <= 90) || ((int)n >= 97 && (int)n <= 122)){  
 int ans = (int)n, exp = (int)n;  
 while(--p) ans \*= exp;  
 return ans;  
 }else{  
 cout << "forbidden action: ";  
 return 0;  
 }  
}  
  
int main() {  
 int exp;  
 short short1;  
 long long1;  
 float float1;  
 double double1;  
 char char1;  
  
 cin >> exp >> short1 >> long1 >> float1 >> double1 >> char1 ;  
 cout << "short: " << power(short1 ,exp) << endl;  
 cout << "long: " << power(long1 ,exp) << endl;  
 cout << "float: " << power(float1,exp) << endl;  
 cout << "double: " << power(double1 ,exp) << endl;  
 cout << "char: " << power(char1 ,exp) << endl;  
  
 return 0;  
}  
  
/\*  
 2 4 100000 3.14 3.141592 A  
 2 4 100000 3.14 3.141592 z  
 3 4 100000 3.14 3.141592 5  
 3 4 100000 3.14 3.141592 z  
 3 4 12345 2.71 3.141592 !  
 \*/

**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++:**

// 1.3  
#include <iostream>  
#include <vector>  
using namespace std;  
  
struct Employee{  
 string fullname;  
 int age;  
 string specialty;  
 float salary;  
};  
  
int main() {  
 int n;  
 cout << "Enter number of factories: " << endl;  
 cin >> n;  
 vector<vector<Employee>> factories;  
 int turnersAmount = 0, locksmithsAmount = 0;  
 for(int i = 0; i < n; i++) {  
  
 cout << "Information for factory #" << i+1 << endl  
 << "Enter factory's employee amount: " << endl;  
 int m;  
 cin >> m;  
 vector<Employee> employees;  
 for(int j = 0; j < m; j++){  
 Employee emp;  
 cout << "Enter " << j+1 << " employee information in next format: fullname, age, specialty, salary: " << endl;  
 cin >> emp.fullname >> emp.age >> emp.specialty >> emp.salary;  
 employees.push\_back(emp);  
 }  
  
 int lCount = 0, tCount = 0;  
 for(auto emp : employees){  
 if(emp.specialty == "turner" || emp.specialty == "Turner") tCount++;  
 else if(emp.specialty == "locksmith" || emp.specialty == "Locksmith") lCount++;  
 }  
 cout << "In factory #" << i+1 << ":" << endl << lCount << " of locksmiths;" << endl << tCount << " of turners." << endl;  
 turnersAmount += tCount;  
 locksmithsAmount += lCount;  
 factories.push\_back(employees);  
 }  
 cout << "Total of locksmiths: " << locksmithsAmount << endl << "Total of turners: " << turnersAmount << endl;  
  
 return 0;  
}  
  
/\*  
2  
5  
aaa 10 turner 100  
bbb 20 turnerr 200  
ccc 30 locksmith 300  
ddd 40 turner 400  
eee 50 locksmith 500  
4  
fff 60 locksmith 600  
ggg 70 locksmith 700  
hhh 80 locksmit 800  
iii 90 turner 900  
 \*/

**Тесты:**

