#include <iostream>

#include <math.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

std::cout << "Привет мир!\n";

// For36 - 38, While1 - 5

// For36 Даны целые положительные числа N и K Найти сумму

// 1^K + 2^K + \_\_\_ + N^K

// Чтобы избежать целочисленного переполнения, вычислять слагаемые

// этой суммы с помощью вещественной переменной и выводить результат как вещественное число

/\*

int N, K;

float count = 0;

cout << "Введите целое положительное число N=";

cin >> N;

cout << "Введите целое положительное число K=";

cin >> K;

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

count += (pow(i, K));

cout << "\nПромежуточный результат = " << count;

}

cout << "\nРезультат выполнения функции =" << count;

\*/

// For37 Дано целое число N (> 0) Найти сумму

// 1^1 + 2^2 + \_\_\_ + N^N

// Чтобы избежать целочисленного переполнения, вычислять слагаемые

// этой суммы с помощью вещественной переменной и выводить результат как вещественное число

/\*

int N;

float count = 0;

cout << "Введите целое положительное число больше нуля N=";

cin >> N;

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

count += pow(i, i);

cout << "\nПромежуточный результат = " << count;

}

cout << "\nРезультат выполнения функции =" << count;

\*/

// For38 Дано целое число N (> 0) Найти сумму

// 1^N + 2^N−1 + \_\_\_ + N^1

// Чтобы избежать целочисленного переполнения, вычислять слагаемые

// этой суммы с помощью вещественной переменной и выводить резуль

// тат как вещественное число

/\*

int N;

float count = 0;

cout << "Введите целое положительное число больше нуля N=";

cin >> N;

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

count += pow(i, N - i + 1);

cout << "\nПромежуточный результат = " << count;

}

cout << "\nРезультат выполнения функции =" << count;

\*/

//While1

// Даны положительные числа A и B(A > B) На отрезке длины A

// размещено максимально возможное количество отрезков длины B(без

// наложений) Не используя операции умножения и деления, найти длину

// незанятой части отрезка A

/\*

int A, B;

float count = 0;

cout << "Введите целое положительное число A=";

cin >> A;

cout << "Введите целое положительное число B которое меньше A =";

cin >> B;

do {

A = A - B;

count++;

cout << "\n №" << count;

} while (A >= B);

cout << "\n Не занятая часть" << A;

\*/

//While2

// Даны положительные числа A и B(A > B)

// На отрезке длины A размещено максимально возможное количество отрезков

// длины B(без наложений) Не используя операции умножения и деления, найти количество

// отрезков B, размещенных на отрезке A

/\*

int A, B;

float count = 0;

cout << "Введите целое положительное число A=";

cin >> A;

cout << "Введите целое положительное число B которое меньше B =";

cin >> B;

do {

A = A - B;

count++;

cout << "\n №" << count;

} while (A >= B);

cout << "\nКоличество отрезков=" << count;

\*/

//While3 Даны целые положительные числа N и K Используя только операции

//сложения и вычитания, найти частное от деления нацело N на K, а также

//остаток от этого деления

/\*

int N, K, temp, count = 0;

cout << "Введите целое положительное число N=";

cin >> N;

cout << "Введите целое положительное число K =";

cin >> K;

while (N >= K) {

temp = N - K;

N = N - K;

count++;

cout << "\nЧастное от деления = " << count;

}

cout << "\nЧастное от деления = " << count;

cout << "\nОстаток от деления = " << temp;

\*/

//While4

//Дано целое число N(> 0) Если оно является степенью числа 3, то

//вывести TRUE, если не является — вывести FALSE

/\*

int N;

cout << "Введите целое положительное число N которое больше нуля =";

cin >> N;

while (N > 2) {

N = N - 3;

cout << "\n №" << N;

}

bool Res = (N == 0);

cout << "\nРезультат = " << (Res ? "True" : "False") << endl;

\*/

//While5 Дано целое число N (> 0), являющееся некоторой степенью числа 2:

//N = 2^K Найти целое число K — показатель этой степени

/\*

int N, count = 0;

cout << "Введите целое N больше нуля, являющееся степенью двойки =";

cin >> N;

while (N > 2) {

N = N / 2;

count++;

cout << "\n" << N;

}

cout << "\nРезультат =" << count;

\*/

// 2 вариант - задачи c четными номерами из диапазона While6-19, For40

//For40 Даны целые числа A и B (A < B) Вывести все целые числа от A до B

//включительно; при этом число A должно выводиться 1 раз, число A + 1

//должно выводиться 2 раза и

/\*

int A, B;

int count = 0;

cout << "Введите целое число A=";

cin >> A;

cout << "Введите целое число B которое больше A =";

cin >> B;

for (A; A <= B + 1; A++) {

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

cout << "\nЧисло " << count << " =" << A;

}

count++;

}

\*/

//While6 Дано целое число N (> 0) Найти двойной факториал N:

//N!! = N·(N−2)·(N−4)·\_\_\_

//(последний сомножитель равен 2, если N — четное, и 1, если

// N — нечетное) Чтобы избежать целочисленного переполнения,

// вычислять это произведение с помощью вещественной переменной и

// вывести его как вещественное число

/\*

int N;

cout << "Введите целое число N больше нуля = ";

cin >> N;

float sum = N;

while (N > 2) {

N = N - 2;

sum \*= N;

cout << "\n Текущий шаг = " << N;

cout << "\n Результат вычисления = " << sum;

}

\*/

//While8 Дано целое число N (> 0) Найти наибольшее целое число K, квадрат

//которого не превосходит N : K^2 ≤ N Функцию извлечения квадратного

//корня не использовать

/\*

int N, K = 1;

cout << "Введите целое число N больше нуля = ";

cin >> N;

while (K \* K <= N) {

cout << "\nВычисляемое в данный моент число K=" << K;

K++;

}

cout << "\n Результат = " << K - 1;

\*/

//While10 Дано целое число N (> 1) Найти наибольшее целое число K, при

//котором выполняется неравенство 3^K < N

/\*

int N, K = 0;

cout << "Введите целое число N больше единицы = ";

cin >> N;

while (pow(3, K) < N) {

K++;

cout << "\nВычисляемое в данный моент число K=" << K;

}

cout << "\n Результат = " << K - 1;

\*/

// While12 Дано целое число N(> 1) Вывести наибольшее из целых чисел K,

// для которых сумма 1 + 2 + \_\_\_ + K будет меньше или равна N, и саму эту сумму

/\*

int N, K = 0, sum = 0;

cout << "Введите целое число N больше единицы = ";

cin >> N;

while (sum + K <= N) {

K++;

sum = sum + K;

cout << "\n Промежуточное значение = " << sum;

}

cout << "\n Результат = " << K ;

\*/

//While14 Дано число A (> 1) Вывести наибольшее из целых чисел K, для

//которых сумма 1 + 1 / 2 + \_\_\_ + 1 / K будет меньше A, и саму эту сумму

/\*

int A, K = 0, sum = 0;

cout << "Введите целое число A больше единицы = ";

cin >> A;

while (sum + K < A) {

K++;

sum = sum + K;

cout << "\n Промежуточное значение = " << sum;

}

cout << "\n Результат = " << K ;

\*/

//While16 Спортсмен-лыжник начал тренировки, пробежав в первый день

//10 км Каждый следующий день он увеличивал длину пробега на P процентов

//от пробега предыдущего дня(P — вещественное, 0 < P < 50)

//По данному P определить, после какого дня суммарный пробег лыжника за

//все дни превысит 200 км, и вывести найденное количество дней K(целое)

//и суммарный пробег S(вещественное число)

/\*

float P, next\_day = 0, count = 0, last\_day = 10, max = 200;

int K = 0;

cout << "Введите вещественное число P меньше пятидесяти, но больше нуля=";

cin >> P;

P = P \* 0.01;

cout << "P=" << P;

while (count < max) {

next\_day = (last\_day \* P) + last\_day;

count += last\_day;

cout << "Новый день = " << next\_day;

last\_day = next\_day;

cout << "Пройдено расстояние =" << count << endl;

K++;

}

cout << "\n Затрачено дней K= " << K;

cout << "\n Пройдено расстояния S= " << count;

\*/

//While18

//Дано целое число N (> 0)

//Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр

int N;

cout << "Введите вещественное число N больше нуля=";

cin >> N;

while (N > 0) {

cout << "\nПромежуточное значение =" << N % 10;

N /= 10;

}

return 0;

}