#include <iostream>

using namespace std;

int main() {

int number, steps = 0;

// Вводимо натуральне число

cout << "Enter a natural number: ";

cin >> number;

// Перевіряємо, що число позитивне і не дорівнює нулю

if (number <= 0) {

cout << "The number must be positive and non-zero." << endl;

return 1; // Завершуємо програму, якщо введено некоректне число

}

// Виконуємо цикл до тих пір, поки number не стане рівним 1

while (number != 1) {

cout << number << endl; // Виводимо поточне значення number

// Якщо число парне, ділимо його на 2

if (number % 2 == 0) {

number /= 2;

} else { // Інакше, якщо число непарне, множимо його на 3 і додаємо 1

number = 3 \* number + 1;

}

steps++; // Збільшуємо лічильник кроків

}

// Виводимо останнє значення 1 і кількість кроків

cout << number << endl;

cout << "Steps = " << steps << endl;

return 0;

}

#include <iostream>

using namespace std;

int main(void) {

double pi4 = 0.0; // Змінна для зберігання суми дробів

long n; // Змінна для кількості ітерацій

// Запитуємо у користувача кількість ітерацій

cout << "Number of iterations? ";

cin >> n;

// Обчислюємо значення Pi/4 за формулою Лейбница

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

// Додаємо або віднімаємо дроби, використовуючи (-1)^i

pi4 += (i % 2 == 0 ? 1.0 : -1.0) / (2.0 \* i + 1);

}

// Виводимо значення Pi

cout.precision(20); // Встановлюємо точність виводу

cout << "Pi = " << (pi4 \* 4.0) << endl; // Помножуємо на 4, щоб отримати значення Pi

return 0;

}

#include <iostream>

using namespace std;

void horizontal\_line(int n){

cout << '+' ;

for(int i = 0; i < n-2; i++){

cout << "-";

}

cout << '+' << endl;

}

void vertical\_line(int n){

for(int i = 0; i < n-2; i++) {

cout << '|';

for(int j = 0; j < n-2; j++){

cout << " ";

}

cout << "|" << endl;

}

}

int main(void) {

int n;

cout << "Введіть значення ";

cin >> n;

if(n>1 && n<80){

horizontal\_line(n);

vertical\_line(n);

horizontal\_line(n);

}

else if (n<2){

cout << "Size must be greater than 1" << endl;

}

else{

cout << "Size must be less than 80" << endl;

}

}