#include <stdio.h>

int main(){

float V=0;

float V1=0;

float V2=0;

float t=0;

printf("V=");

scanf("%f", &V);

printf("V1=");

scanf("%f", &V1);

printf("V2=");

scanf("%f", &V2);

if (V1>V2){t=V/(V1-V2);}

else {printf("error");}

printf("t=%f\n", t);

return 0;

}

#include <stdio.h>

int main()

{

float s = 0;

float n = 1;

float a = 1;

float b = 1;

float eps;

printf("eps=");

scanf("%f", &eps);

while (n > eps)

{

s = s + n;

a = a + 2;

b = b + 3;

n = n \* (a / b);

}

printf("s=%f\n", s);

return 0;

}

#include <stdio.h>

int main() {

char str[100]; // Объявление массива символов для хранения введенной строки

int count = 0; // Счетчик для подсчета количества серий цифр и букв

int length = 0; // Переменная для хранения длины текущей серии

int prevLength = 0; // Переменная для хранения длины предыдущей серии

int i = 0; // Индекс для обхода массива

int c; // Переменная для хранения текущего введенного символа

printf("Enter a string: "); // Вывод запроса на ввод строки

while ((c = getchar()) != '\n' && c != EOF) { // Чтение строки до символа новой строки или конца файла

str[i++] = c;

}

str[i] = '\0'; // Добавление завершающего нулевого символа в строку

for (i = 0; str[i] != '\0'; i++) { // Обход строки для анализа символов

if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= '0' && str[i] <= '9')) { // Проверка, является ли символ буквой или цифрой

length++;

} else {

if (length > 0) { // Если серия не пустая

if (count == 0) { // Первая серия

prevLength = length;

} else {

if (length != prevLength) { // Проверка на совпадение длины текущей серии с предыдущей

printf("No\n");

return 0;

}

}

count++;

length = 0;

}

}

}

if (length > 0 && length != prevLength) { // Проверка последней серии, если она не завершена символом новой строки

printf("No\n");

return 0;

}

printf("yes\n");

return 0;

}

#include <stdio.h> // Подключение стандартной библиотеки ввода-вывода

int main(void) {

char line[1000]; // Объявление массива символов для хранения строки

fgets(line, sizeof(line), stdin); // Чтение строки из стандартного ввода

remove\_numbers(line); // Вызов функции для удаления цифр из строки

printf("%s", line); // Вывод отредактированной строки на экран

return 0; // Завершение работы программы

}

// Функция для удаления цифр из строки

void remove\_numbers(char \*str) {

char \*src = str, \*dst = str; // Инициализация указателей на начало строки

while (\*src) { // Цикл, пока не достигнут конец строки

if (\*src < '0' || \*src > '9') { // Если символ не является цифрой

\*dst++ = \*src; // Копирование символа в новый буфер

}

src++; // Переход к следующему символу в исходной строке

}

\*dst = '\0'; // Завершение строки нулевым символом

}

#include <stdio.h>

// Определение константы N, равной 10

#define N 10

int main(void) {

// Объявление массива x размером N и переменной i

int x[N];

int i;

// Объявление переменных для хранения сумм четных и нечетных элементов

int sum\_even = 0, sum\_odd = 0;

// Цикл для ввода элементов массива и подсчета суммы четных и нечетных элементов

for (i = 0; i < N; i++) {

scanf("%d", &x[i]);

if (i % 2 == 0) {

// Если индекс элемента четный, добавляем его к sum\_even

sum\_even += x[i];

} else {

// Иначе добавляем его к sum\_odd

sum\_odd += x[i];

}

}

// Цикл для проверки условий и обнуления элементов массива

for (i = 0; i < N; i++) {

// Если индекс четный и сумма четных элементов меньше суммы нечетных,

// или индекс нечетный и сумма четных элементов больше или равна сумме нечетных,

// то обнуляем элемент массива

if ((i % 2 == 0 && sum\_even < sum\_odd) || (i % 2 != 0 && sum\_even >= sum\_odd)) {

x[i] = 0;

}

}

// Цикл для вывода элементов массива

for (i = 0; i < N; i++) {

printf("%d ", x[i]);

}

printf("\n");

// Возвращаем 0, что означает успешное завершение программы

return 0;

}

#include <stdio.h>

#define K 3 // Определение количества строк

#define N 4 // Определение количества столбцов

int main(void) {

int x[K][N]; // Объявление двумерного массива размером K на N

int aver\_global; // Переменная для хранения глобального среднего арифметического

int aver\_row; // Переменная для хранения среднего арифметического каждой строки

int i, j, k; // Объявление переменных-счетчиков

// Цикл для ввода элементов массива

for (i = 0; i < K; i++) {

for (j = 0; j < N; j++) {

scanf("%d", &x[i][j]); // Ввод элемента массива

}

}

// Вычисление глобального среднего арифметического значения

aver\_global = 0;

for (i = 0; i < K; i++) {

for (j = 0; j < N; j++) {

aver\_global += x[i][j]; // Суммирование всех элементов массива

}

}

aver\_global /= (K \* N); // Расчет среднего арифметического

// Цикл для обнуления строк, среднее арифметическое которых меньше глобального

for (i = 0; i < K; i++) {

aver\_row = 0;

for (j = 0; j < N; j++) {

aver\_row += x[i][j]; // Суммирование элементов текущей строки

}

aver\_row /= N; // Расчет среднего арифметического текущей строки

if (aver\_row < aver\_global) { // Проверка условия

for (k = 0; k < N; k++) {

x[i][k] = 0; // Обнуление элементов строки

}

}

}

// Вывод обновленного массива

for (i = 0; i < K; i++) {

for (j = 0; j < N; j++) {

printf("%4d ", x[i][j]); // Вывод элемента массива

}

printf("\n"); // Переход на новую строку после вывода строки массива

}

return 0;

}

#include <stdio.h>

unsigned long reverseBitsWithMask(unsigned long n, unsigned long mask) {

unsigned long reversed = 0;

for (int i = 0; i < 64; ++i) {

if ((n & (mask >> i)) > 0) {

reversed |= (1 << (63 - i));

}

}

return reversed;

}

int main() {

unsigned long n, mask;

printf("Введите длинное целое число: ");

scanf("%llu", &n);

printf("Введите маску: ");

scanf("%llu", &mask);

unsigned long reversed = reverseBitsWithMask(n, mask);

printf("Число с перевернутыми битами в маске: %llu\n", reversed);

return 0;

}