МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе №2**

«Классификация языков программирования»

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ» *ассистент кафедры ЗИ*

*Савицкий Н. А. Исаев Г. А.*

«\_\_» \_\_\_\_\_\_ 2024г«\_\_\_» \_\_\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2024

**Цели и задачи работы**: изучение функций ввода-вывода данных, программирования вычисления значения выражения.

**Задание к работе**:

1. Реализовать линейный вычислительный процесс. Самостоятельно решить задачу в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования JavaScript, Go (Golang), Swift, С#, Haskell, Ruby, Rust, C++, Python, Java, Kotlin.

2. Реализовать линейный вычислительный процесс любого задания на языке программирования Assembler.

3. Представленные задачи можно реализовать на каждом языке в одной программе с последовательным выполнением.

**Задание 1.6.** Вам дана строка s, содержащая только буквы. Необходимо определить количество гласных и согласных букв, после чего вывести сообщение в соответствии со следующими условиями: если количество гласных> количество согласных, то вывести - “Да”. Если количество гласных <количество согласных, то вывести - “Нет”. Если количество гласных = количество согласных, то вывести - “Одинаково”.

**Задание 2.6.** Учитывая массив nums целых чисел, вернуть, сколько из них содержат нечетное количество цифр.

**Задание 3.1.** Необходимо каждое из этих чисел перевернуть и вывести без ведущих нулей.

**Методика выполнения работы:**

1. Определить типы используемых в программе данных.

2. Описать переменные.

3. Написать функции ввода-вывода.

4. Разработать алгоритм решения задачи по индивидуальному заданию.

5. Написать и отладить программу с вводом-выводом информации

6. Протестировать работу программы на различных исходных данных.

7. Изменить формат вывода, проверить работу программы при другом формате вывода.

**Программы решения задачи на языке C++ с их результатами работы.**

### Задание 1.6.

#include <iostream>

#include <vector>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

string glasnye = "yaoeiu";

cout << "Введите строку: ";

string s;

char c;

cin.get(c);

while (c != '\n') {

if (c != ' ') {

s += c;

}

cin.get(c);

}

// Проверяем, что введенная строка не пустая

if (s.empty()) {

cout << "Ошибка: введена пустая строка." << endl;

return 1; // Возвращаем ненулевой код, чтобы указать на ошибку

}

int quantityGlas = 0;

for (int i = 0; i < s.size(); i++) {

for (int j = 0; j < glasnye.size(); j++) {

if (s[i] == glasnye[j]) {

quantityGlas++;

}

}

}

int soglas = s.size() - quantityGlas;

if (quantityGlas > soglas) {

cout << "Да";

} else if (quantityGlas == soglas) {

cout << "Одинаково";

} else {

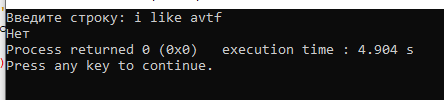
cout << "Нет";

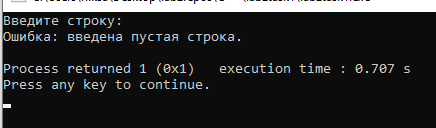
}

return 0;

}

### Результат выполнения

****

****

### Задание 2.6.

#include <iostream>

#include <vector>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

vector<int> nums;

int n;

cout << "Введите количество элементов: ";

cin >> n;

// Проверяем, что введенное количество элементов неотрицательное число

if (n <= 0) {

cout << "Ошибка: введено некорректное количество элементов." << endl;

return 1; // Возвращаем ненулевой код, чтобы указать на ошибку

}

cout << "Введите числа: " << endl;;

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

int num;

cin >> num;

nums.push\_back(num);

}

int j = 0;

for (auto num : nums) {

int digitCount = 0;

int temp = num;

// Подсчитываем количество цифр в числе

while (temp != 0) {

temp /= 10;

digitCount++;

}

// Если количество цифр нечетное, увеличиваем счетчик

if (digitCount % 2 != 0) {

j++;

}

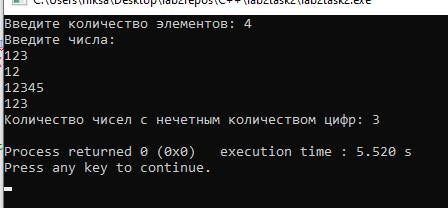
}

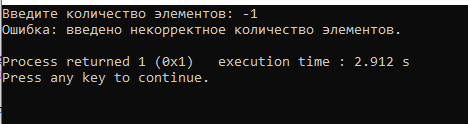
cout << "Количество чисел с нечетным количеством цифр: " << j << endl;

return 0;

}

### Результат выполнения





### Задание 3.6.

#include <iostream>

#include <vector>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

vector<int> nums;

int n;

cout << "Введите количество символов: ";

cin >> n;

// Проверяем, что введенное количество символов неотрицательное число

if (n <= 0) {

cout << "Ошибка: введено некорректное количество символов." << endl;

return 1; // Возвращаем ненулевой код, чтобы указать на ошибку

}

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

int elem;

cout << "Элемент: ";

cin >> elem;

int revElem = 0;

while (elem > 0) {

revElem = revElem \* 10 + elem % 10;

elem = elem / 10;

}

nums.push\_back(revElem);

}

for (auto elems : nums) {

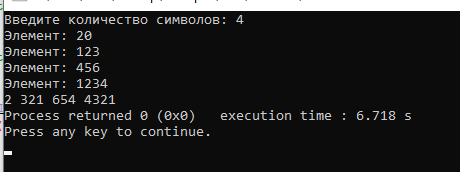
cout << elems << " ";

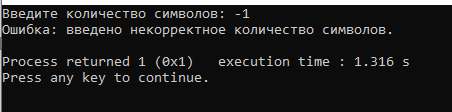
}

return 0;

}

### Результат выполнения



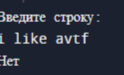


### Программы решения задачи на языке Rust с их результатами работы.

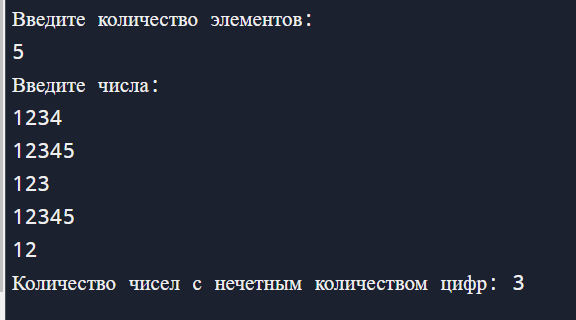
### Задание 1.6.

use std::io::{self, BufRead}; // Подключаем необходимые модули  
  
fn main() {  
 // Создаем строку с гласными буквами  
 let vowels = "aeiou";  
  
 // Ввод строки от пользователя без пробелов  
 println!("Введите строку:");  
  
 let mut input\_string = String::new(); // Создаем переменную для хранения введенной строки  
 io::stdin().lock().read\_line(&mut input\_string).expect("Ошибка при считывании строки");  
 input\_string.retain(|c| c != ' ' && c != '\n'); // Удаляем пробелы и символы перевода строк  
  
 let mut quantityGlas = 0; // Переменная для хранения количества гласных букв  
  
 // Подсчет количества гласных букв и согласных букв в строке  
 for c in input\_string.chars() {  
 if c.is\_alphabetic() { // Проверка, является ли символ буквой  
 for v in vowels.chars() {  
 // Проверяем, является ли символ гласной буквой  
 if c.to\_ascii\_lowercase() == v {  
 quantityGlas += 1; // Увеличиваем количество гласных букв  
 break; // Прерываем цикл поиска гласных букв  
 }  
 }  
 } else {  
 println!("Некорректный ввод!"); // Выводим ошибку при вводе не буквенных символов  
 return; // Прекращаем выполнение программы  
 }  
 }  
  
 let quantitySoglas= input\_string.len() - quantityGlas; // Вычисляем количество согласных букв  
  
 // Сравниваем количество гласных и согласных букв и выводим результат  
 if quantityGlas > 0 {  
 if quantityGlas > quantitySoglas {  
 println!("Да");  
 } else if quantityGlas == quantitySoglas {  
 println!("Одинаково");  
 } else {  
 println!("Нет");  
 }  
 }  
}

### Результат выполнения



### Задание 2.6.



### Результат выполнения

use std::io;

fn main() {

// Задаем региональные настройки для вывода сообщений на русском языке

println!("Введите количество элементов:");

// Создаем переменную для хранения количества элементов

let mut n = String::new();

// Считываем ввод пользователя в переменную n

io::stdin().read\_line(&mut n).expect("Ошибка при чтении строки");

// Преобразуем строку в число типа i32

let n: i32 = n.trim().parse().expect("Введите целое число");

println!("Введите числа:");

// Создаем вектор для хранения чисел

let mut nums: Vec<i32> = Vec::new();

// Считываем n чисел и добавляем их в вектор

for \_ in 0..n {

let mut num = String::new();

io::stdin().read\_line(&mut num).expect("Ошибка при чтении строки");

let num: i32 = num.trim().parse().expect("Введите целое число");

nums.push(num);

}

// Переменная для подсчета чисел с нечетным количеством цифр

let mut count = 0;

// Проходим по каждому числу в векторе

for num in &nums {

// Преобразуем число в строку и убираем пробелы

let num\_str = num.to\_string().replace(" ", "");

// Подсчитываем количество цифр в числе

let digit\_count = num\_str.chars().filter(|c| c.is\_digit(10)).count();

// Если количество цифр нечетное, увеличиваем счетчик

if digit\_count % 2 != 0 {

count += 1;

}

}

// Выводим количество чисел с нечетным количеством цифр

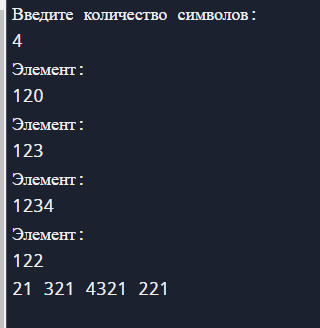
println!("Количество чисел с нечетным количеством цифр: {}", count);

}

### Задание 3.6.

use std::io::{self, BufRead};  
  
fn main() {  
 // Создаем вектор для хранения перевернутых чисел  
 let mut nums: Vec<i32> = Vec::new();  
  
 // Запрашиваем количество чисел, которое пользователь хочет ввести  
 println!("Введите количество символов: ");  
 let mut input = String::new();  
 io::stdin().lock().read\_line(&mut input).expect("Ошибка при считывании строки");  
  
 // Удаляем пробелы из вводимой строки  
 let input = input.replace(" ", "");  
  
 // Преобразуем введенное значение в число  
 let n: i32 = input.trim().parse().expect("Ошибка при преобразовании числа");  
  
 // Проверяем, что введено корректное количество символов  
 if n <= 0 {  
 println!("Ошибка: введено некорректное количество символов.");  
 return; // Завершаем программу  
 }  
  
 // Ввод и обработка каждого элемента  
 for \_ in 0..n {  
 println!("Элемент: ");  
 let mut input\_elem = String::new();  
 io::stdin().lock().read\_line(&mut input\_elem).expect("Ошибка при считывании строки");  
  
 // Преобразуем введенный элемент в число  
 let elem: i32 = input\_elem.trim().parse().expect("Ошибка при преобразовании числа");  
  
 let mut rev\_elem = 0;  
 // Переворачиваем число  
 let mut num = elem;  
 while num > 0 {  
 rev\_elem = rev\_elem \* 10 + num % 10;  
 num = num / 10;  
 }  
   
 nums.push(rev\_elem); // Добавляем перевернутое число в вектор  
 }  
  
 // Выводим перевернутые числа  
 for elem in nums {  
 print!("{} ", elem);  
 }  
}

### Результат выполнения



### Программы решения задачи на языке C# с их результатами работы.

### Задание 1.6.

using System;

class Program

{

static void Main()

{

// Устанавливаем кодовую страницу консоли для поддержки русского языка

Console.OutputEncoding = System.Text.Encoding.UTF8;

// Создаем строку с гласными буквами

string glasnye = "yaoeiu";

// Ввод строки от пользователя и удаление пробелов

Console.Write("Введите строку: ");

string s = Console.ReadLine().Replace(" ", "");

int quantityGlas = 0;

// Подсчет количества гласных и согласных букв в строке

foreach (char ch in s)

{

if (Char.IsLetter(ch)) // Проверка, является ли символ буквой

{

foreach (char glas in glasnye)

{

if (Char.ToLower(ch) == glas) // Сравниваем символ с гласными

{

quantityGlas++; // Увеличиваем счетчик гласных

}

}

}

else

{

Console.WriteLine("Некорректный ввод!"); // Выводим сообщение об ошибке

return; // Завершаем программу

}

}

int soglas = s.Length - quantityGlas; // Вычисляем количество согласных

// Сравнение количества гласных и согласных букв и вывод результата

if (quantityGlas != 0)

{

if (quantityGlas > soglas)

{

Console.WriteLine("Да");

}

else if (quantityGlas == soglas)

{

Console.WriteLine("Одинаково");

}

else

{

Console.WriteLine("Нет");

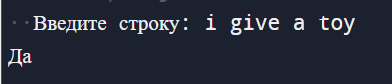
}

}

}

}

### Результат выполнения



### Задание 2.6.

using System;

using System.Collections.Generic;

class Program

{

static void Main(string[] args)

{

// Устанавливаем культуру для отображения сообщений на английском языке

System.Threading.Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo("en-US");

// Запрашиваем количество элементов у пользователя

Console.Write("Enter the number of elements: ");

int n = int.Parse(Console.ReadLine());

// Создаем список для хранения чисел

List<int> nums = new List<int>();

// Запрашиваем числа у пользователя

Console.WriteLine("Enter the numbers:");

for (int i = 0; i < n; i++)

{

int num = int.Parse(Console.ReadLine());

nums.Add(num);

}

// Переменная для подсчета чисел с нечетным количеством цифр

int count = 0;

// Проходим по каждому числу в списке

foreach (int num in nums)

{

int digitCount = 0;

int temp = num;

// Подсчитываем количество цифр в числе

while (temp != 0)

{

temp /= 10;

digitCount++;

}

// Если количество цифр нечетное, увеличиваем счетчик

if (digitCount % 2 != 0)

{

count++;

}

}

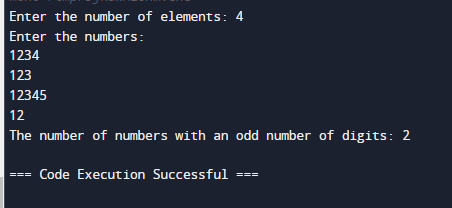
// Выводим количество чисел с нечетным количеством цифр

Console.WriteLine("The number of numbers with an odd number of digits: " + count);

}

}

### Результат выполнения



### Задание 3.6.

using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

// Устанавливаем кодовую страницу консоли для поддержки русского языка

Console.OutputEncoding = System.Text.Encoding.UTF8;

// Создаем список для хранения перевернутых чисел

List<int> nums = new List<int>();

// Вводим количество элементов от пользователя

Console.WriteLine("Введите количество элементов: ");

int n;

if (!int.TryParse(Console.ReadLine(), out n) || n <= 0)

{

// Если введено некорректное количество элементов, выводим ошибку и завершаем программу

Console.WriteLine("Ошибка: введено некорректное количество элементов.");

return;

}

// Считываем числа и переворачиваем их

for (int i = 0; i < n; i++)

{

Console.Write("Элемент: ");

if (!int.TryParse(Console.ReadLine(), out int elem))

{

// Если введено некорректное число, выводим ошибку и завершаем программу

Console.WriteLine("Ошибка: введено некорректное число.");

return;

}

int revElem = 0;

// Переворачиваем число

while (elem > 0)

{

revElem = revElem \* 10 + elem % 10;

elem = elem / 10;

}

// Добавляем перевернутое число в список

nums.Add(revElem);

}

// Выводим перевернутые числа

Console.WriteLine("Перевернутые числа:");

foreach (int elem in nums)

{

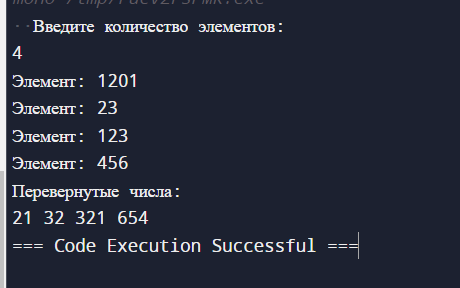
Console.Write(elem + " ");

}

}

}

### Результат выполнения



### Программы решения задачи на языке Java с их результатами работы.

### Задание 1.6.

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

// Создаем строку с гласными буквами

String glasnye = "yaoeiu";

// Создаем объект Scanner для ввода данных

Scanner scanner = new Scanner(System.in);

// Ввод строки от пользователя

System.out.print("Введите строку: ");

String s = "";

String input = scanner.nextLine().replaceAll(" ", "");

// Удаление пробелов из вводимой строки

for (char c : input.toCharArray()) {

if (c != ' ') {

s += c;

}

}

int quantityGlas = 0;

// Подсчет количества гласных букв и согласных букв в строке

for (int i = 0; i < s.length(); i++) {

char ch = s.charAt(i);

if (Character.isLetter(ch)) { // Проверка, является ли символ буквой

for (int j = 0; j < glasnye.length(); j++) {

if (Character.toLowerCase(ch) == glasnye.charAt(j)) {

quantityGlas++;

}

}

} else {

System.out.println("Некорректный ввод!");

return;

}

}

int soglas = s.length() - quantityGlas;

// Сравнение количества гласных и согласных букв

if (quantityGlas != 0) {

if (quantityGlas > soglas) {

System.out.println("Да");

} else if (quantityGlas == soglas) {

System.out.println("Одинаково");

} else {

System.out.println("Нет");

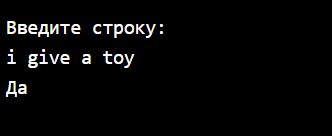
}

}

}

}

### Результат выполнения



### Задание 2.6.

import java.util.Scanner;

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

// Создание объекта Scanner для считывания ввода с консоли

Scanner scanner = new Scanner(System.in);

// Создание списка для хранения чисел

ArrayList<Integer> nums = new ArrayList<>();

// Ввод пользователем количества элементов

System.out.print("Введите количество элементов: ");

int n = scanner.nextInt();

// Ввод пользователем чисел и добавление их в список

System.out.println("Введите числа:");

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

int num = scanner.nextInt();

nums.add(num);

}

// Переменная для подсчета количества чисел с нечетным количеством цифр

int count = 0;

// Перебор всех чисел в списке

for (int num : nums) {

int digitCount = 0;

int temp = num;

// Подсчет количества цифр в текущем числе

while (temp != 0) {

temp /= 10;

digitCount++;

}

// Если количество цифр нечетное, увеличиваем счетчик

if (digitCount % 2 != 0) {

count++;

}

}

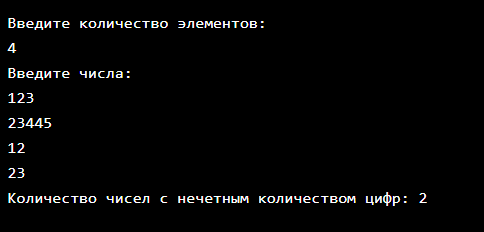
// Вывод количества чисел с нечетным количеством цифр

System.out.println("Количество чисел с нечетным количеством цифр: " + count);

}

}

### Результат выполнения



### Задание 3.6.

import java.util.Scanner;

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Integer> nums = new ArrayList<>(); // создаем массив для хранения перевернутых чисел

System.out.println("Введите количество символов: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Ошибка: введено некорректное количество символов.");

return; // завершаем программу

}

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

System.out.print("Элемент: ");

int elem = Integer.parseInt(scanner.next().replaceAll("\\s+", "")); // считываем элемент и удаляем пробелы

int revElem = 0;

// переворачиваем число

while (elem > 0) {

revElem = revElem \* 10 + elem % 10;

elem = elem / 10;

}

nums.add(revElem); // добавляем перевернутое число в ArrayList

}

// выводим перевернутые числа

for (int elems : nums) {

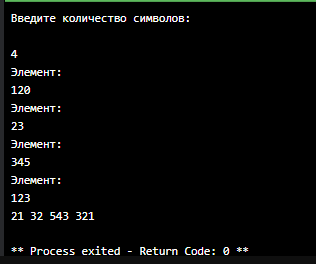
System.out.print(elems + " ");

}

}

}

### Результат выполнения



### Программы решения задачи на языке JavaScript с их результатами работы.

### Задание 1.6.

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

// Создаем строку с гласными буквами

const glasnye = "yaoeiu";

// Запрашиваем ввод строки от пользователя

rl.question('Введите строку: ', (input) => {

let s = input.replace(/\s/g, ''); // Удаляем пробелы из введенной строки

let quantityGlas = 0;

// Подсчет количества гласных букв и согласных букв в строке

for (let i = 0; i < s.length; i++) {

if (s[i].match(/[a-zA-Z]/)) { // Проверка, является ли символ буквой

for (let j = 0; j < glasnye.length; j++) {

if (s[i].toLowerCase() === glasnye[j]) {

quantityGlas++;

}

}

} else {

console.log("Некорректный ввод!");

break;

}

}

let soglas = s.length - quantityGlas;

// Сравнение количества гласных и согласных букв

if (quantityGlas > 0) {

if (quantityGlas > soglas) {

console.log("Да");

} else if (quantityGlas === soglas) {

console.log("Одинаково");

} else {

console.log("Нет");

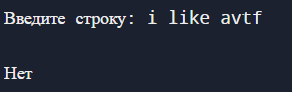
}

}

rl.close();

});

### Результат выполнения



### Задание 2.6.

// Подключаем модуль для ввода-вывода

const readline = require('readline');

// Создаем интерфейс для чтения из стандартного потока ввода

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

// Функция для подсчета количества цифр в числе

function countDigits(num) {

let digitCount = 0;

while (num !== 0) {

num = Math.floor(num / 10);

digitCount++;

}

return digitCount;

}

// Функция для подсчета количества чисел с нечетным количеством цифр

function countOddDigitNumbers(nums) {

let count = 0;

for (let num of nums) {

let digitCount = countDigits(num);

// Если количество цифр нечетное, увеличиваем счетчик

if (digitCount % 2 !== 0) {

count++;

}

}

return count;

}

// Запрашиваем количество элементов

rl.question('Введите количество элементов: ', (n) => {

let nums = [];

// Запрашиваем числа и добавляем их в массив

console.log('Введите числа:');

rl.on('line', (input) => {

nums.push(parseInt(input));

// Если в массиве уже n чисел, останавливаем ввод

if (nums.length === parseInt(n)) {

rl.close();

}

});

rl.on('close', () => {

// Подсчитываем количество чисел с нечетным количеством цифр

let count = countOddDigitNumbers(nums);

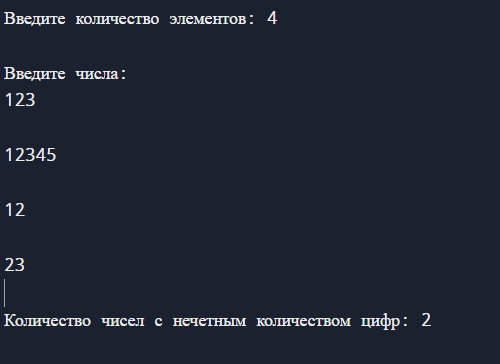
console.log('Количество чисел с нечетным количеством цифр: ' + count);

process.exit(0);

});

});

### Результат выполнения



### Задание 3.6.

const readline = require('readline'); // Подключаем модуль readline для взаимодействия с пользователем через консоль

const rl = readline.createInterface({ // Создаем интерфейс readline

input: process.stdin, // Устанавливаем ввод из стандартного потока ввода

output: process.stdout // Устанавливаем вывод в стандартный поток вывода

});

let nums = []; // Создаем массив для хранения перевернутых чисел

rl.question('Введите количество символов: ', (n) => { // Задаем вопрос пользователю и ждем ответа

n = parseInt(n.trim()); // Преобразуем введенное значение в целое число

if (n <= 0 || isNaN(n)) { // Проверяем, является ли введенное значение некорректным

console.log("Ошибка: введено некорректное количество символов."); // Выводим сообщение об ошибке

rl.close(); // Закрываем интерфейс readline

return; // Завершаем программу, если введено некорректное количество символов

}

console.log("Элемент: "); // Выводим приглашение для ввода элементов

let count = 0; // Инициализируем счетчик

rl.on('line', (input) => { // Слушаем событие ввода пользователя

console.log("Элемент: "); // Выводим приглашение для ввода следующего элемента

let elem = parseInt(input.trim()); // Преобразуем введенное значение в целое число

let revElem = 0; // Инициализируем переменную для хранения перевернутого числа

// Переворачиваем число

while (elem > 0) {

revElem = revElem \* 10 + elem % 10; // Переворачиваем число путем добавления последней цифры

elem = Math.floor(elem / 10); // Удаляем последнюю цифру числа

}

nums.push(revElem); // Добавляем перевернутое число в массив

count++; // Увеличиваем счетчик

if (count === n) { // Проверяем, достигнуто ли заданное количество элементов

rl.close(); // Закрываем интерфейс readline

}

});

rl.on('close', () => { // Слушаем событие закрытия интерфейса readline

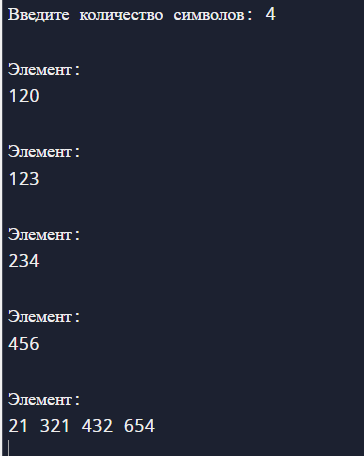
// Выводим перевернутые числа

console.log(nums.join(' ')); // Выводим перевернутые числа через пробел

});

});

### Результат выполнения



### Программы решения задачи на языке Go с их результатами работы.

### Задание 1.6.

package main

import (

"fmt"

"unicode"

)

func main() {

// Создаем строку с гласными буквами

glasnye := "yaoeiu"

// Ввод строки от пользователя

fmt.Print("Введите строку: ")

var s string

var c rune

for {

// Считываем один символ за раз, пока не нажата клавиша Enter

fmt.Scanf("%c", &c)

if c == '\n' {

break

}

if c != ' ' {

// Добавляем символ к строке s, если он не является пробелом

s += string(c)

}

}

quantityGlas := 0

// Подсчет количества гласных букв и согласных букв в строке

for \_, char := range s {

if unicode.IsLetter(char) { // Проверка, является ли символ буквой

for \_, glas := range glasnye {

if unicode.ToLower(char) == glas {

quantityGlas++

}

}

} else {

fmt.Println("Некорректный ввод!")

return

}

}

soglas := len(s) - quantityGlas

// Сравнение количества гласных и согласных букв

if quantityGlas != 0 {

if quantityGlas > soglas {

fmt.Println("Да")

} else if quantityGlas == soglas {

fmt.Println("Одинаково")

} else {

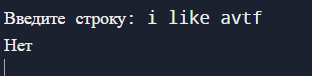
fmt.Println("Нет")

}

}

}

### Результат выполнения



### Задание 2.6.

package main

import (

"fmt"

)

func main() {

// Запрашиваем количество элементов у пользователя

fmt.Print("Введите количество элементов: ")

var n int

fmt.Scan(&n)

// Создаем слайс для хранения чисел

nums := make([]int, 0)

// Запрашиваем числа у пользователя

fmt.Println("Введите числа:")

for i := 0; i < n; i++ {

var num int

fmt.Scan(&num)

nums = append(nums, num)

}

// Переменная для подсчета чисел с нечетным количеством цифр

count := 0

// Проходим по каждому числу в слайсе

for \_, num := range nums {

digitCount := 0

temp := num

// Подсчитываем количество цифр в числе

for temp != 0 {

temp /= 10

digitCount++

}

// Если количество цифр нечетное, увеличиваем счетчик

if digitCount%2 != 0 {

count++

}

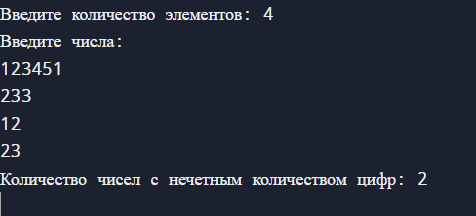
}

// Выводим количество чисел с нечетным количеством цифр

fmt.Println("Количество чисел с нечетным количеством цифр:", count)

}

### Результат выполнения



### Задание 3.6.

package main

import (

"fmt" // Импортируем пакет fmt для ввода-вывода

)

func main() {

// Создаем срез для хранения перевернутых чисел

var nums []int

var n int

fmt.Print("Введите количество символов: ") // Выводим приглашение для ввода количества чисел

fmt.Scan(&n) // Считываем количество чисел, введенных пользователем

if n <= 0 {

fmt.Println("Ошибка: введено некорректное количество символов.") // Если введено некорректное количество символов, выводим ошибку

return // Завершаем программу с ошибкой

}

// Считываем числа и переворачиваем их

for i := 0; i < n; i++ { // Цикл для считывания чисел и переворачивания их

var elem int

fmt.Print("Элемент: ") // Выводим приглашение для ввода следующего числа

fmt.Scan(&elem) // Считываем число, введенное пользователем

revElem := 0

// Переворачиваем число

for elem > 0 {

revElem = revElem\*10 + elem%10 // Получаем последнюю цифру числа и добавляем её в реверсивное число

elem = elem / 10 // Убираем последнюю цифру из числа

}

nums = append(nums, revElem) // Добавляем перевернутое число в срез

}

// Выводим перевернутые числа

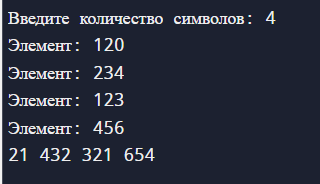
for \_, elem := range nums { // Цикл для вывода перевернутых чисел

fmt.Printf("%d ", elem) // Выводим каждое перевернутое число через пробел

}

}

### Результат выполнения



### Программы решения задачи на языке Python с их результатами работы.

### Задание 1.6.

glasnye = "yaoeiu"

s = input("Введите строку: ")

quantityGlas = 0

for char in s:

if char.lower() in glasnye:

quantityGlas += 1

soglas = len(s) - quantityGlas

if quantityGlas > soglas:

print("Да")

elif quantityGlas == soglas:

print("Одинаково")

else:

print("Нет")

### Результат выполнения



### Задание 2.6.

# Создаем пустой список для хранения чисел

nums = []

# Вводим количество элементов

n = int(input("Введите количество элементов: "))

print("Введите числа:")

# Читаем n чисел, введенных пользователем

for i in range(n):

num = int(input())

nums.append(num)

# Переменная для подсчета чисел с нечетным количеством цифр

count = 0

# Проходим по каждому числу в списке

for num in nums:

# Считаем количество цифр в числе

digitCount = len(str(num))

# Если количество цифр нечетное, увеличиваем счетчик

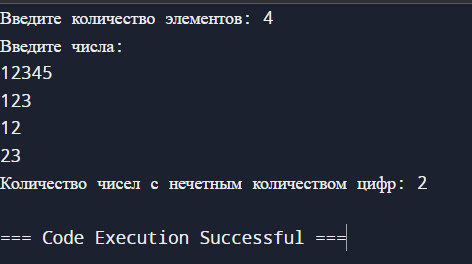
if digitCount % 2 != 0:

count += 1

# Выводим количество чисел с нечетным количеством цифр

print("Количество чисел с нечетным количеством цифр:", count)

### Результат выполнения



### Задание 3.6.

print("Введите количество символов:")

n = int(input())

nums = []

for i in range(n):

print("Элемент:")

elem = int(input())

revElem = 0

while elem > 0:

revElem = revElem \* 10 + elem % 10

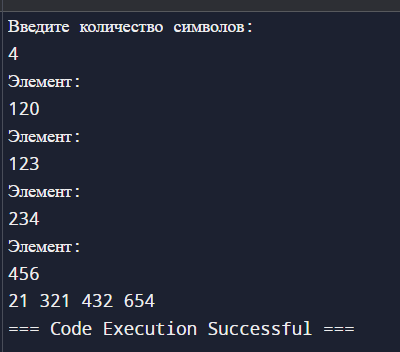
elem = elem // 10

nums.append(revElem)

for elem in nums:

print(elem, end=" ")

### Результат выполнения



### Программы решения задачи на языке Ruby с их результатами работы.

### Задание 1.6.

glasnye = "yaoeiu"

print "Введите строку: "

s = gets.chomp.delete(' ')

# Проверяем, что введенная строка не пустая

if s.empty?

puts "Ошибка: введена пустая строка."

exit 1 # Завершаем программу, указывая на ошибку

end

quantity\_glas = 0

s.each\_char do |c|

quantity\_glas += 1 if glasnye.include?(c.downcase)

end

soglas = s.size - quantity\_glas

if quantity\_glas > soglas

puts "Да"

elsif quantity\_glas == soglas

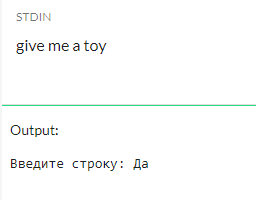
puts "Одинаково"

else

puts "Нет"

end

### Результат выполнения



### Задание 2.6.

# Устанавливаем региональные настройки на русский язык

# (необходимо для корректного отображения сообщений на кириллице)

Encoding.default\_external = Encoding::UTF\_8

Encoding.default\_internal = Encoding::UTF\_8

# Создаем пустой массив для хранения чисел

nums = []

# Запрашиваем количество элементов у пользователя

puts "Введите количество элементов:"

n = gets.chomp.to\_i

# Выводим приглашение для ввода чисел

puts "Введите числа:"

# Считываем n чисел, введенных пользователем, и добавляем их в массив

n.times do

num = gets.chomp.to\_i

nums << num

end

# Переменная для подсчета чисел с нечетным количеством цифр

count = 0

# Проходим по каждому числу в массиве

nums.each do |num|

digit\_count = 0

temp = num

# Подсчитываем количество цифр в числе

while temp != 0

temp /= 10

digit\_count += 1

end

# Если количество цифр нечетное, увеличиваем счетчик

if digit\_count.odd?

count += 1

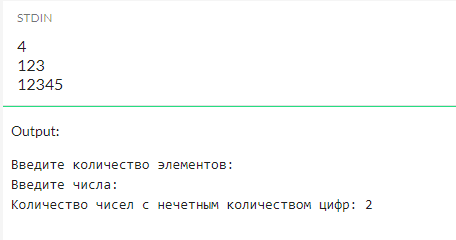
end

end

# Выводим количество чисел с нечетным количеством цифр

puts "Количество чисел с нечетным количеством цифр: #{count}"

### Результат выполнения



### Задание 3.6.

# Создаем массив для хранения перевернутых чисел

nums = []

# Ввод количества символов

puts "Введите количество символов: "

n = gets.chomp.to\_i

if n <= 0

puts "Ошибка: введено некорректное количество символов."

exit(1) # завершаем программу с ошибкой

end

# Цикл ввода чисел, их переворачивания и добавления в массив

n.times do

print "Элемент: "

elem = gets.chomp.to\_i

rev\_elem = 0

# Переворачиваем число

while elem > 0

rev\_elem = rev\_elem \* 10 + elem % 10

elem = elem / 10

end

nums.push(rev\_elem) # добавляем перевернутое число в массив

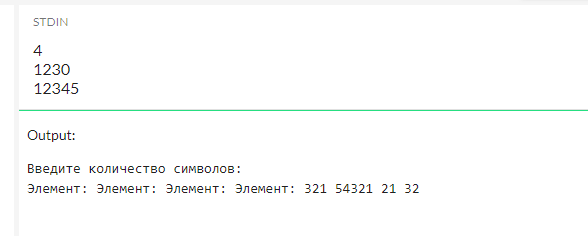
end

# Вывод перевернутых чисел

nums.each { |elem| print "#{elem} " }

puts

### Результат выполнения



### Программы решения задачи на языке Swift с их результатами работы.

### Задание 1.6.

// Создаем строку с гласными буквами

let glasnye = "yaoeiu"

// Ввод строки от пользователя

print("Введите строку: ", terminator: "")

var s = ""

if let input = readLine() {

for c in input {

if c != " " {

s += String(c)

}

}

}

var quantityGlas = 0

var invalidInput = false // Флаг для обозначения некорректного ввода

// Подсчет количества гласных букв и согласных букв в строке

for char in s {

if char.isLetter { // Проверка, является ли символ буквой

if let lowerChar = String(char).lowercased().first { // Получаем символ в нижнем регистре

if glasnye.contains(lowerChar) {

quantityGlas += 1

}

}

} else {

invalidInput = true

print("Некорректный ввод!")

break

}

}

if !invalidInput {

let soglas = s.count - quantityGlas

// Сравнение количества гласных и согласных букв

if quantityGlas != 0 {

if quantityGlas > soglas {

print("Да")

} else if quantityGlas == soglas {

print("Одинаково")

} else {

print("Нет")

}

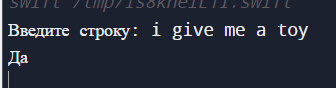
} else {

print("Нет гласных букв в строке.")

}

}

### Результат выполнения



### Задание 2.6.

import Foundation

// Устанавливаем региональные настройки для вывода сообщений на русском языке

setlocale(LC\_ALL, "Russian")

// Создаем пустой массив для хранения чисел

var nums = [Int]()

// Запрашиваем количество элементов у пользователя

print("Введите количество элементов:")

// Считываем ввод пользователя и преобразуем в Int

if let input = readLine(), let n = Int(input) {

// Выводим приглашение для ввода чисел

print("Введите числа:")

// Считываем n чисел, введенных пользователем, и добавляем их в массив

for \_ in 0..<n {

if let num = readLine(), let number = Int(num) {

nums.append(number)

}

}

}

// Переменная для подсчета чисел с нечетным количеством цифр

var count = 0

// Проходим по каждому числу в массиве

for num in nums {

var digitCount = 0

var temp = num

// Подсчитываем количество цифр в числе

while temp != 0 {

temp /= 10

digitCount += 1

}

// Если количество цифр нечетное, увеличиваем счетчик

if digitCount % 2 != 0 {

count += 1

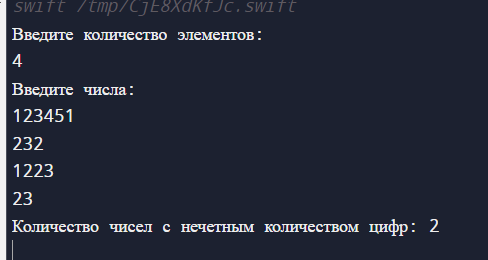
}

}

// Выводим количество чисел с нечетным количеством цифр

print("Количество чисел с нечетным количеством цифр: \(count)")

### Результат выполнения



### Задание 3.6.

import Foundation

// Объявление массива для хранения чисел

var nums: [Int] = []

// Вывод запроса на ввод количества символов

print("Введите количество символов:")

// Считывание введенной строки

if let input = readLine(), let n = Int(input) {

// Проверка, что введенное значение больше нуля

if n <= 0 {

print("Ошибка: введено некорректное количество символов.")

exit(1)

}

// Цикл для считывания элементов заданное количество раз

for \_ in 0..<n {

print("Элемент:")

// Считывание введенной строки и преобразование в число

if let input = readLine(), let elem = Int(input) {

// Переворачивание числа и добавление его в массив

let revElem = Int(String(String(elem).reversed())) ?? 0

nums.append(revElem)

}

}

// Вывод элементов массива через пробел

for elem in nums {

print(elem, terminator: " ")

}

} else {

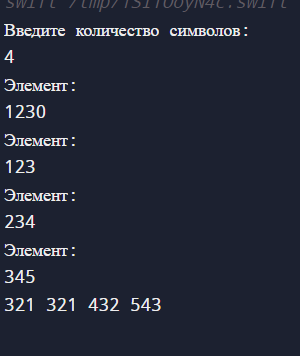
// Вывод сообщения об ошибке при некорректном вводе

print("Ошибка: введено некорректное количество символов.")

exit(1)

}

### Результат выполнения



### Программы решения задачи на языке Haskell с их результатами работы.

### Задание 2.6.

import Data.Char (digitToInt)

-- Функция для подсчета количества цифр в числе

количество\_цифр :: Int -> Int

количество\_цифр 0 = 0

количество\_цифр n = 1 + количество\_цифр (n `div` 10)

-- Функция для проверки числа на нечетное количество цифр

нечетное\_количество\_цифр :: Int -> Bool

нечетное\_количество\_цифр n = количество\_цифр n `mod` 2 /= 0

-- Функция для подсчета чисел с нечетным количеством цифр в списке

подсчет\_нечетных\_цифр :: [Int] -> Int

подсчет\_нечетных\_цифр = length . filter нечетное\_количество\_цифр

main = do

-- Запрашиваем количество элементов у пользователя

putStrLn "Введите количество элементов:"

-- Считываем ввод пользователя и преобразуем его в число

n <- fmap read getLine :: IO Int

-- Запрашиваем числа у пользователя

putStrLn "Введите числа:"

-- Считываем n чисел, введенных пользователем

числа <- sequence $ replicate n (fmap read getLine :: IO Int)

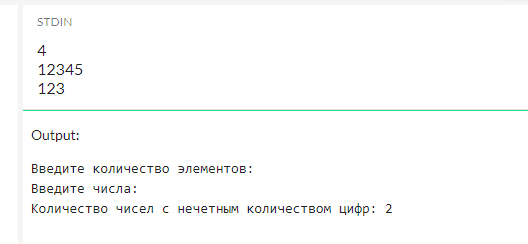
-- Подсчитываем количество чисел с нечетным количеством цифр

let результат = подсчет\_нечетных\_цифр числа

-- Выводим результат

putStrLn $ "Количество чисел с нечетным количеством цифр: " ++ show результат

### Результат выполнения



### Программы решения задачи на языке Assembler с их результатами работы.

### Задание 2.6.

%include "io64.inc"

section .data

nums: times 255 dw 0 ; Буфер для чисел

n: equ $ - nums

section .bss

input\_num resw 1 ; Буфер для одного вводимого числа

section .text

global main

main:

mov rbp, rsp ; for correct debugging

; Инициализация индекса и количества элементов

xor rsi, rsi ; индекс для доступа к nums обнуляем перед использованием

; Считываем количество элементов в массиве

mov rdi, input\_num

GET\_DEC 8, input\_num

NEWLINE

movzx rcx, word [input\_num] ; количество элементов для ввода

push rcx ; сохраняем количество элементов в массиве - для дальнейших вычислений

input\_loop:

mov rdi, input\_num ; адрес буфера для ввода числа

call GET\_DEC 8, input\_num ; ввод числа

movzx rax, word [input\_num] ; загружаем введенное число

mov [nums + rsi], ax ; сохраняем число в массив

add rsi, 2 ; следующий индекс в массиве

dec rcx ; уменьшаем счетчик

jnz input\_loop ; переходим в начало цикла, если rcx не равно 0

; Обработка чисел из массива

xor rsi, rsi ; сброс индекса для обработки массива

xor rbp, rbp ; счетчик чисел с нечетным количеством цифр

pop rcx ; восстанавливаем количество элементов в мвссиве

numsloop:

movzx rax, word [nums + rsi] ; Загружаем число из массива

xor rdi, rdi ; Обнуляем счетчик цифр в числе

test rax, rax

jz zero\_digit ; если число равно нулю, обрабатываем особо

nnn:

inc rdi

mov edx, 0 ; обнуляем rdx перед делением, так как div использует dx:ax

mov bx, 10

div bx ; rax = rax / 10, rdx = rax % 10

test rax, rax

jne nnn ; продолжаем, пока не обработаем все цифры

zero\_digit:

test rdi, 1 ; проверяем, нечетное ли количество цифр

jz Chet ; если четное, переходим к Chet

inc rbp ; инкрементируем счетчик нечетных чисел

Chet:

add rsi, 2 ; перемещаемся к следующему элементу массива

loop numsloop ; продолжаем цикл по массиву

; Вывод результата

mov rdi, rbp ; Подготавливаем значение для вывода

NEWLINE

PRINT\_DEC 8, rdi ; Выводим результат

ret

### Результат выполнения

