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. «АБ-121», «АВТФ» *доцент кафедры ЗИ*

*Новиков Ян Валерьевич Ершов И. А.*

«\_\_\_» \_\_\_\_\_\_ 2022г«\_\_\_» \_\_\_\_\_\_ 2022 г.

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

(подпись) (подпись)

Новосибирск 2022

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

**Задание к работе:**

Реализовать линейный вычислительный процесс. Самостоятельно решить задачу в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования C++, Python, Haskell (только одно задание), C#, Go, Ruby (только одно задание), Java.

Реализовать линейный вычислительный процесс любого задания на языке программирования Assembler.

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

***Задания на разных языках программирования***

Текст заданий:

**Задание 1**: Даны координаты двух противоположных вершин прямоугольника: (x1, y1), (x2, y2). Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.

**Задание 2**: У Вас на даче растут цветы, которым требуется постоянный полив. Цветы расположены в ряд и нумеруются от 0 до n-1 слева направо. Рядом течет река, в которой вы можете наполнить свою лейку при x = -1.

Примечание: каждый цветок нуждается в поливе определенным количество воды.

Алгоритм полива цветов:

• Направление: слева направо.

• Если после полива i-ого цветка не хватает полива для (i+1) цветка, то нужно вернуться и набрать полную лейку.

• Режим экономии: лейки можно наполнять в реке только тогда, когда это требуется.

Для решения задачи необходимо предусмотреть вариант нахождения рядом с рекой в позиции -1. Для перемещения можно шагать по горизонтальной оси на 1.

Необходимо определить число шагов для полива всех цветов для заданного количества и потребности каждого цветка.

**Задание 3**: Определить, сколько чисел из последовательности равны предыдущему числу.

**С++**

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

float SquareArea(pair<float, float> a, pair<float, float> b)

{

float c = fabs(a.first - b.first);

float d = fabs(a.second - b.second);

return c \* d;

}

float SquarePerim(pair<float, float> a, pair<float, float> b)

{

float c = fabs(a.first - b.first);

float d = fabs(a.second - b.second);

return (c + d) \* 2;

}

int Number3(vector <int> a)

{

int h = 0;

for (int i = 1; i < a.size(); i++)

{

if (a.at(i) == a.at(i - 1))

h += 1;

}

return h;

}

int Flowers(vector<int> a, int dlin, int lei)

{

int b = lei;

int steps = 1;

for (int i = 0; i < dlin - 1; i++)

{

if (a.at(i + 1) > b - a.at(i))

{

steps += (i + 1) + i + 2;

b = lei;

}

else

{

b -= a.at(i);

steps += 1;

}

}

return steps;

}

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "First number" << endl;

int x1, y1, x2, y2;

cout << "Введите координаты первой точки" << endl;

cin >> x1 >> y1;

pair<float, float> a(x1, y1);

cout << "Введите координаты второй точки" << endl;

cin >> x2 >> y2;

pair<float, float> b(x2, y2);

float result1;

cout << "Площадь равна ";

result1 = SquareArea(a, b);

cout << result1 << endl;

cout << "Периметр равен ";

result1 = SquarePerim(a, b);

cout << result1 << endl;

cout << "Second number" << endl;

int result2, numfl, lei;

cout << "Введите количество цветков" << endl;

cin >> numfl;

vector<int> potrfl;

cout << "Последовательно введите потребность каждого цветка" << endl;

for (int i = 0, a; i < numfl; i++)

{

cin >> a;

potrfl.push\_back(a);

}

cout << "Введите вместимость лейки" << endl;

cin >> lei;

result2 = Flowers(potrfl, numfl, lei);

cout << "Для полива потребуется " << result2 << " шагов" << endl;

cout << "Third number" << endl;

int kolnum;

cout << "Введите количество цифр в ряду" << endl;

cin >> kolnum;

vector<int> riad;

cout << "Последовательно введите каждый элемент ряда" << endl;

for (int i = 0, a; i < kolnum; i++)

{

cin >> a;

riad.push\_back(a);

}

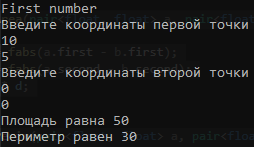
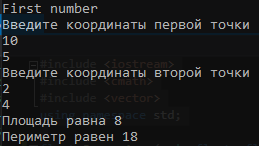
int result3 = Number3(riad);

cout << "Кол-во цифр, удовлетворяющих условию " << result3 << " штук(и)" << endl;

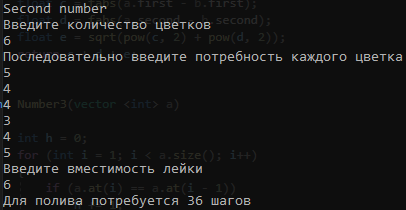
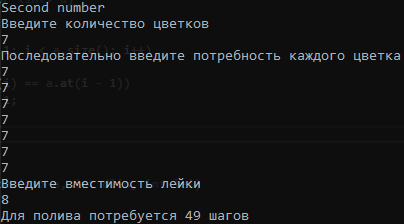
}

*Тесты:*

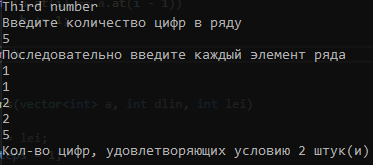
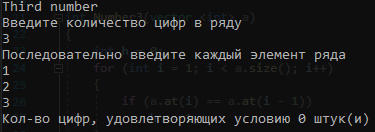
Задание 1:



Задание 2:



Задание 3:



**Python**

import math #математические функции

def ploshad (a, b): #задание функции

c = abs((a[0] - b[0]) \* (a[1] - b[1])) #нахождение площади

print(c)

def perimetr (a, b):

c = (abs(a[0] - b[0]) + abs(a[1] - b[1])) \* 2 #нахождение периметра

print(c)

def flowers (potr, dlin, lei):

b = lei #вместимость лейки

steps = 1 #шаги

for i in range(0, dlin - 1): #бегаем по массиву

if potr[i + 1] > (b - potr[i]):

steps += (i + 1) + i + 2 #шаги при нехватке воды

b = lei

else:

b -= potr[i] #при достатке воды

steps += 1

print(steps)

def num3(a):

h = 0

for i in range(1, len(a)): #бежим по массиву и ищем подходящие под условие элементы ряда

if a[i] == a[i - 1]:

h += 1

print(h)

print("Num1") #вывод строки в консоль

a = [0, 0] # массивы с координатами точек

b = [10, 5]

print("Площадь треугольника равна ")

ploshad(a, b) # вывод площади

print("Периметр треугольника равен ")

perimetr(a, b) #вывод периметра

print("Num2")

potr = [7, 7, 7, 7, 7, 7, 7] #массив с потребностями цветков

print("Кол-во шагов равняется ")

flowers(potr, 7, 8) #вывод кол-ва шагов

print("Num3")

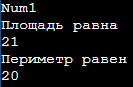
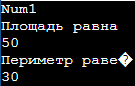
riad = [1, 2, 6, 8, 9, 9] #ряд чисел

print("Кол-во цифр, подходящих под условие, равно ")

num3(riad) # вывод кол-во цифр, подходящих под условие

Задание 1:



Задание 2:









Задание 3:



**C#**

using System;

class Program {

static int num3(int [] riad, int dlin){ // статистическая функция для третьего номера

int h = 0; // объявление и инициализация переменной

for (int i = 1; i < dlin; i++) // бежим по массиву

{

if (riad[i] == riad[i - 1]) // проверка условия

h += 1;

}

return h;

}

static int flowers(int [] potrfl, int kolfl, int lei){ // функция для второго задания

int b = lei; // кол-во воды в лейке

int steps = 1; // шаги

for (int i = 0; i < kolfl - 1; i++) // шаги при нехватке

{

if (potrfl[i + 1] > b - potrfl[i])

{

steps += (i + 1) + i + 2;

b = lei;

}

else // шаги при достатке

{

b -= potrfl[i];

steps += 1;

}

}

return steps; // возвращаем значение шагов

}

static void Main(string[] args) { // функция main

Console.WriteLine("Number 1\n"); // вывод строки и переход на новую

Console.WriteLine("Enter the coordinates of the first point\n"); //Вывод на консоль

double x1 = Convert.ToDouble(Console.ReadLine()); // ввод данных

double y1 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Enter the coordinates of the second point\n");

double x2 = Convert.ToDouble(Console.ReadLine());

double y2 = Convert.ToDouble(Console.ReadLine());

double resultPlosh = Math.Abs((x1 - x2) \* (y1 - y2));

Console.WriteLine("The area of the triangle is " + resultPlosh + "\n"); // площадь треугольника

double resultPerim = (Math.Abs(x1 - x2) + Math.Abs(y1 - y2)) \* 2;

Console.WriteLine("The perimeter of the triangle is " + resultPerim + "\n"); // периметр треугольника

Console.WriteLine("Number 2\n");

Console.WriteLine("Enter the number of flowers\n");

int kolfl = Convert.ToInt32(Console.ReadLine());

int [] potrfl = new int[kolfl];

Console.WriteLine("Consistently enter the need of each flower\n");

for(int i = 0, a; i < kolfl; i++ ) // заполнение массива потребностей цветков

{

a = Convert.ToInt32(Console.ReadLine());

potrfl[i] = a;

}

Console.WriteLine("Enter the volume of the watering can\n");

int lei = Convert.ToInt32(Console.ReadLine());

Console.Write("Needed " + flowers(potrfl, kolfl, lei) + " steps\n"); // вывод ответа на 2 задание

Console.WriteLine("Number 3\n");

Console.WriteLine("Enter the row length\n");

int dlin = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Consistently enter the numbers of row\n");

int [] riad = new int[dlin];

for(int i = 0, a; i < dlin; i++ ) // заполнение ряда чисел

{

a = Convert.ToInt32(Console.ReadLine());

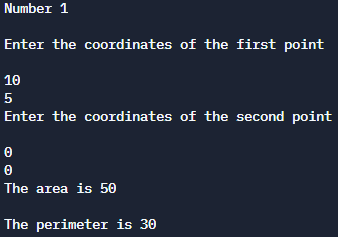
riad[i] = a;

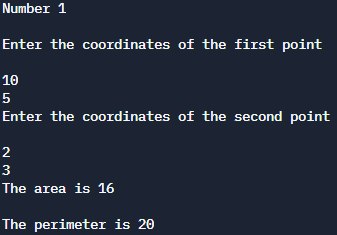
}

Console.Write("Numbers that fit the condition, " + num3(riad, dlin) + " \n"); // ответ на 3 задание

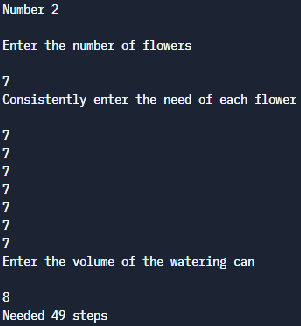
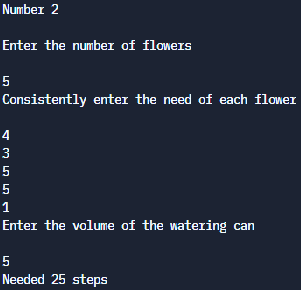
}

}

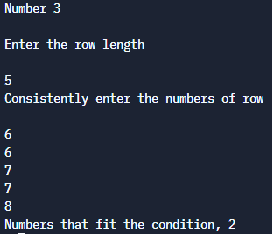
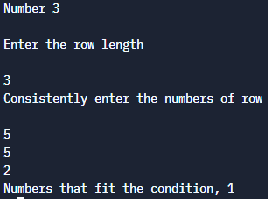
Задание 1:



Задание 2:



Задание 3:

**Haskell**

f x1 y1 x2 y2 = abs (x1 - x2) \* abs (y1 - y2) --функция для площади треугольника

f1 x1 y1 x2 y2 = (abs (x1 - x2) + abs (y1 - y2)) \* 2 --функция нахождения периметра треугольника

main = do

print ("Area is") --вывод строки

print (f 5 2 1 1) --вывод площади

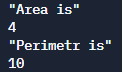
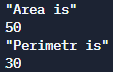
print ("Perimetr is")

print (f1 5 2 1 1) --вывод периметра

Задание 1:





**Ruby**

x1 = 5 #инициализация переменных(координат)

y1 = 3

x2 = 1

y2 = 1

area = (x1 - x2).abs \* (y1 - y2).abs #площадь

puts ("Area = " + area.to\_s) #вывод площади и строки в консоль

perimetr = ((x1 - x2).abs + (y1 - y2).abs) \* 2 #периметр

puts ("Perimetr = " + perimetr.to\_s) #вывод периметра

Задание 1:



**Java**

import java.util.Scanner; // для ввода данных

public class Main { // класс

public static int Num3(int [] riad, int dlin){ //функция для 3 номера

int h = 0; // объявление и инициализация переменной

for (int i = 1; i < dlin; i++) // бежим по массиву

{

if (riad[i] == riad[i - 1]) // проверяем условие

h += 1;

}

return h; // возвращаем значение

}

public static double Area(double x1, double y1, double x2, double y2 ){ // функция нахождения площади

double b = Math.abs(x1 - x2) \* Math.abs(y1 - y2) / 2; // переменная с площадью

return b;

}

public static double Perimetr(double x1, double y1, double x2, double y2 ){ // функция для нахождения периметра

double b = (Math.abs(x1 - x2) + Math.abs(y1 - y2)) \* 2; // переменная с периметром

return b;

}

public static int Flowers(int kolfl, int[] potrfl, int lei){ // функция для 2 задания

int b = lei; // кол-во воды в лейке

int steps = 1; // шаги

for (int i = 0; i < kolfl - 1; i++) // бежим по массиву

{

if (potrfl[i + 1] > b - potrfl[i]) // не хватает на след

{

steps += (i + 1) + i + 2; //считаем шаги

b = lei;

}

else // хватает на след

{

b -= potrfl[i];

steps += 1;

}

}

return steps;

}

public static void main(String args[]) { // функция main

System.out.println("Number 1"); // вывод строки в консоль

Scanner kord = new Scanner(System.in); // объявление сканнера

double x1, y1, x2, y2; // объявление переменных

System.out.println("Enter the coordinates of the first point");

x1 = kord.nextDouble(); // ввод данных типа double в переменную x1

y1 = kord.nextDouble();

System.out.println("Enter the coordinates of the second point");

x2 = kord.nextDouble();

y2 = kord.nextDouble();

System.out.println("Area = " + Area(x1, y1, x2, y2)); // вывод в консоль площади

System.out.println("Perimetr = " + Perimetr(x1, y1, x2, y2)); // вывод периметра

System.out.println("Number 2");

Scanner flow = new Scanner(System.in); // объявляем новый сканнер

System.out.println("Enter the number of flowers");

int kolfl = flow.nextInt();

int [] potrfl = new int[kolfl];

System.out.println("Consistently enter the need of each flower");

for(int i = 0, a; i < kolfl; i++ )

{

a = flow.nextInt();

potrfl[i] = a; // заполнение массива потребностями цветов

}

System.out.println("Enter the volume of the watering can");

int lei = flow.nextInt(); // вместимость лейки

System.out.println("Needed " + Flowers(kolfl, potrfl, lei) + " steps"); // вывод кол-ва шагов

System.out.println("Number 3");

Scanner numthree = new Scanner(System.in); // объявление нового сканнера

System.out.println("Enter the number of numbers");

int dlin = numthree.nextInt();

int [] riad = new int[dlin];

System.out.println("Enter each number of row");

for(int i = 0, a; i < dlin; i++ )

{

a = numthree.nextInt();

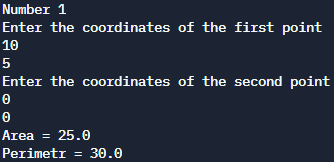
riad[i] = a; // заполнение ряда числами

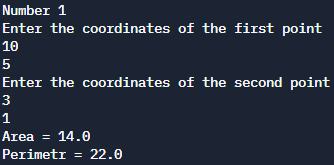
}

System.out.println("The number of numbers that match the condition " + Num3(riad, dlin)); // вывод ответа на 3 задание

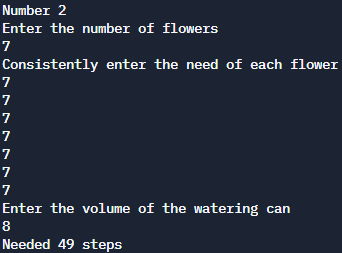
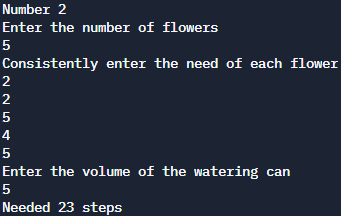
}

}

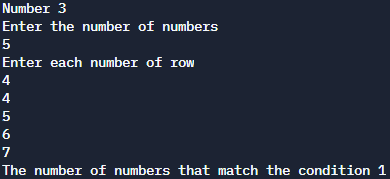
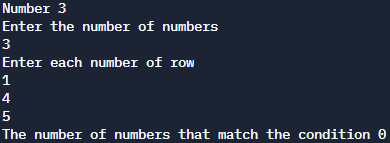
Задание 1:



Задание 2:



Задание 3:



**Go**

package main // пакет main

import "fmt" // библиотека для вывода

import "math" // библиотека мат.функций

func Num3(riad[100] int, dlin int) int{ // функция для 3 номера

var h int = 0 // объявление и инициализация переменной

var i int

for i = 1; i < dlin; i++{ // шагаем по массиву

if riad[i] == riad[i - 1]{ // проверка условия

h += 1

}

}

return h; // возврат значения функцией

}

func Flowers(potrfl[100] int, kolfl int, lei int) int{ // функция для второго задания

var b int = lei // кол-во воды в лейке

var steps int = 1 // шаги

var i int

for i = 0; i < (kolfl - 1); i++{ // шагаем по массиву

if potrfl[i + 1] > (b - potrfl[i]){ // если не хватает на след

steps += (i + 1) + i + 2 // треб шагов

b = lei

}else{ // хватает на след

b -= potrfl[i]

steps += 1

}

}

return steps;

}

func Area(x1 float64, y1 float64, x2 float64, y2 float64) float64{ // функция площади

var a float64 = math.Abs(x1 - x2) \* math.Abs(y1 - y2) // площадь

return a

}

func Perim(x1 float64, y1 float64, x2 float64, y2 float64) float64{ // функция периметра

var a float64 = (math.Abs(x1 - x2) + math.Abs(y1 - y2)) \* 2 // периметр

return a

}

func main() {

var x1, y1, x2, y2 float64

fmt.Println("Number 1") // вывод сообщения в консоль

fmt.Println("Enter the coordinates of the first point")

fmt.Scanf("%float64", &x1) // ввод значения переменной x1 типа float64

fmt.Scanf("%float64", &y1)

fmt.Println("Enter the coordinates of the second point")

fmt.Scanf("%float64", &x2)

fmt.Scanf("%float64", &y2)

fmt.Println("Area = ", Area(x1, y1, x2, y2)) // вывод площади

fmt.Println("Perimetr = ", Perim(x1, y1, x2, y2)) // вывод периметра

fmt.Println("Number 2")

var potrfl[100] int

var kolfl, lei int

fmt.Println("Enter the number of flowers")

fmt.Scanf("%d\n", &kolfl)

fmt.Println("Consistently enter the need of each flower")

var i, a int

for i = 0; i < kolfl; i++{ // цикл заполнения массива потребностей цветков

fmt.Scanf("%d\n", &a)

potrfl[i] = a

}

fmt.Println("Enter the volume of the watering can")

fmt.Scanf("%d\n", &lei)

fmt.Println("Needed", Flowers(potrfl, kolfl, lei), "steps") // вывод необходимого кол-ва шагов

fmt.Println("Number 3")

var riad[100] int

var dlin int

fmt.Println("Enter the number of numbers")

fmt.Scanf("%d\n", &dlin)

fmt.Println("Enter each number of row") // заполнение ряда числами

for i = 0; i < dlin; i++{

fmt.Scanf("%d\n", &a)

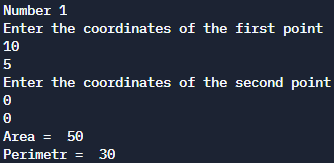
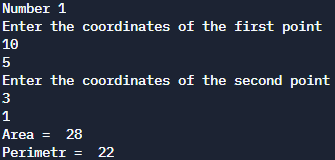
riad[i] = a

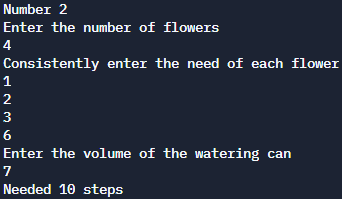
}

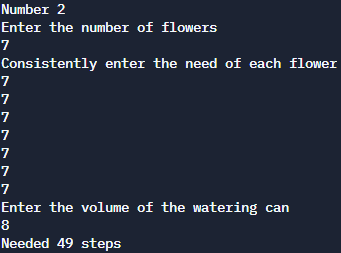
fmt.Println("The number of numbers that match the condition", Num3(riad, dlin)) // вывод ответа на 3 задание

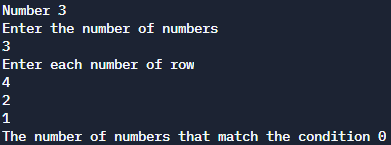
}

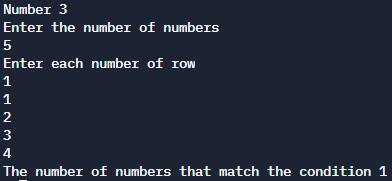
Задание 1:

Задание 2:



Задание 3:



**Assembler(NASM)**

section .text ;Сегмент Code

global \_start ;нужно объявить для использования gcc

\_start: ;говорим линкеру о точке входа

mov eax,'4' ; переносим значение ‘4’ в eax

sub eax, '0'

mov ebx, '1'

sub ebx, '0'

sub eax, ebx

add eax, '0'

mov [razny], eax

mov ecx,msg1 ;сообщение для записи

mov edx, len1 ;длина сообщения

mov ebx,1 ;файловый дескриптор (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov ecx,razny

mov edx, 1

mov ebx,1 ;файловый дескриптор (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov eax,'6'

sub eax, '0'

mov ebx, '4'

sub ebx, '0' ; вычитаем ascii '0' для конвертации её в десятичную цифру

sub eax, ebx

add eax, '0' ;обратно в ascii

add ebx, '0'

mov [raznx], eax

mov ecx,msg2

mov edx, len2

mov ebx,1 ;файловый дескриптор (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov ecx,raznx

mov edx, 1

mov ebx,1 ;файловый дескриптор (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov eax,[raznx] ;вычисляем произведение высоты и ширины

sub eax, '0'

mov ebx, [razny]

sub ebx, '0' ; разность

mul ebx ; произведение

add eax, '0'

add ebx, '0'

mov [proizv], eax

mov ecx,msg3

mov edx, len3

mov ebx,1 ;дескриптор файла (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov ecx,proizv

mov edx, 1

mov ebx,1 ;файловый дескриптор (stdout)

mov eax,4 ;номер системного вызова (sys\_write)

int 0x80 ;вызов ядра

mov eax,1 ;номер системного вызова (sys\_exit)

int 0x80 ;вызов ядра

section .data

msg1 db "Высота равна:", 0xA,0xD

len1 equ $ - msg1 ;длина сообщения

msg2 db 0xA,0xD ,"Ширина равна:", 0xA,0xD

len2 equ $ - msg2

msg3 db 0xA,0xD ,"Площадь равна:", 0xA,0xD

len3 equ $ - msg3

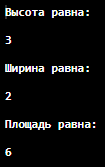
segment .bss ;Неинициализированные данные

razny resb 1

raznx resb 1

proizv resb 1

Задание 1:



**Дополнительные задания**

Задание 1: Составить список из 10 "странных" языков программирования.

**Список странных ЯП**

**1.LolCode**

Как вы уже понимаете из названия, язык явно не претендует на мировую популярность. Язык был разработан еще в 2007 году и в качестве синтаксиса для него был использован сленговый английский и фразы с картинок интернет-мема Lolcat.

**2.YoptaScript**

В отличии от прошлого примера YoptaScript написан нашими ребятами, студентами из петегбурзского университета. YoptaScript – это язык для настоящих гопников, которые хотят из дворов перебраться в элиту программистов.

**3.Chicken**

Если прошлые языки давали нам надежду их освоить, то язык Chicken явно не ставил перед собой такую цель. Ключевых слов здесь не много – только одно. Да, вы правы, это «chicken». В каждой строке содержится определенное количество слов «chicken», и от их числа зависит, какая команда будет выполняться.

**4.Malbolge**

Особенность этого языка в том, что он работает в троичной системе счисления и имеет всего три регистра для указателей и управления данными. Ещё у этого языка нет стандарта синтаксиса, команды могут идти в произвольном порядке. А чтобы было ещё интереснее, после загрузки вся свободная память интерпретатора заполняется с помощью команды crazy: она сравнивает в троичной системе 2 числа по своим правилам и результат отправляется в память.

**5.Piet**

Язык работает так: компилятор оценивает разницу между соседними цветами, чтобы понять, что ему делать дальше. При этом он ещё перебирает оттенки цветов по кругу — так программа всегда знает, к какому цвету нужно перейти дальше.

Кроме оттенков программа сравнивает и яркость цветов: светло-синий и тёмно-синий — это два разных цвета, каждый из которых обрабатывается по-своему.

**6.WhiteSpace**

Суть языка в том, что в качестве символов для написания кода используются лишь непечатаемые конструкции, а именно: пробел, перевод строки и табуляция.

**7.Brain Fuck**

Brain Fuck не столь уж плохой по сравнению с другими. На нем действительно можно написать программу, но ваш мозг при этом будет соответствовать названию языка.

По языку есть четкая, маленькая документация, следуя ей можно написать полноценную программу.

**8.ArnoldC**

Любителей кино порадует язык программирования ArnoldC, составленный из цитат героев Арнольда Шварцнеггера. Не будем углубляться в подробности.

**9.Shakespeare**

Shakespeare — эзотерический язык программирования, разработанный Джоном Аслаудом и Карлом Хассельстромом. Язык Shakespeare призван замаскировать исходный код программы под пьесы Уильяма Шекспира.

Список персонажей в начале программы служит для объявления количества стеков, носящих имена, например, «Ромео» и «Джульетта». Эти герои общаются друг с другом — тем самым совершая операции ввода-вывода, могут задавать вопросы (аналог условного оператора). Части исходного кода на Shakespeare называются Актами (Act), которые разбиваются на Сцены (Scene). Каждые Акт и Сцена пронумерованы римскими цифрами и служат метками для GOTO.

**10.Chef**

Chef — эзотерический язык программирования, разработанный Дэвидом Морган-Маром, программы на котором сходны с кулинарными рецептами. Каждая программа в языке состоит из названия, списка переменных и их значений, списка инструкций. Переменные могут быть названы только названиями основных продуктов питания. Стек, в которые помещаются значения переменных, называется англ. mixing bowl («чаша для смешивания»), а операции для манипуляции с переменными — mix («смешать»), stir («взболтать») и так далее. Язык идеально подходит для домохозяек и для Джеймса Бонда.

Задание 2: Составить таблицу, отображающую хронологию создания ЯП. До 1980 года укрупненными блоками по 10 лет, с 1980 года с инкрементом 1 год.

Ссылка на Google Таблицу: <https://docs.google.com/spreadsheets/d/1Cq0hQNHeCeDAJK0ItZEkVP0q56_IogxMkxKuHyUrv9A/edit?usp=sharing>

**Вывод**

В ходе проделанной работы был реализован линейный вычислительный процесс. Решены задачи в соответствии с индивидуальным вариантом на разных языках программирования, а именно: C++, Python, C#, Go, Haskell, Java, Ruby, Assembler. Также были выполнены дополнительные задания к лабораторной работе и изучена теоретическая часть.