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*

*Вобликов Валерий Евгеньевич Архипова А. Б.*

«\_\_\_» \_\_\_\_\_\_ 20\_\_г«\_\_\_» \_\_\_\_\_\_ 2022 г.

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

(подпись) (подпись)

Новосибирск 2022

**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.

Вариант 5

5. Дана длина ребра куба a. Найти объем куба V = a3 и площадь его поверхности S = 6·a2

Вариант 5

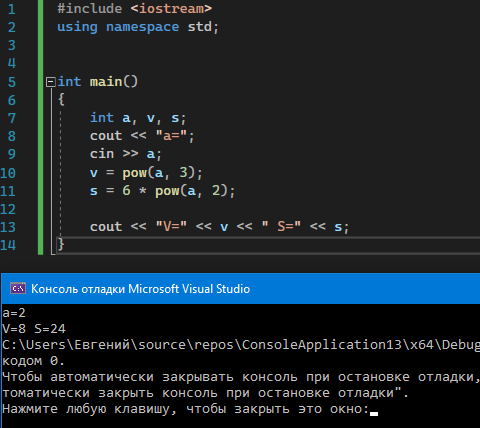
Дан массив строк, где каждое слово может быть записано как с позиций азбуки Морзе. Определить количество уникальных слов на азбуке Морзе при условии, что буквы можно переставлять. Принять длину слова в строке не более 11 символов. Количество слов не ограничено.

Вариант 5

Для каждого числа определить, является ли оно простым (Простыми называются числа, которые делятся ТОЛЬКО на 1 и на самих себя). Пример. Вход: 4 17 32 5 9 Выход: 2.

***C++***

1. **C++**

****

**2)C++**

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

int main()

{

char eng[26] = { 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z' };

string mas[26] = { "ab","baaa", "baba", "baa", "a", "aaba", "bba", "aaaa", "aa", "abbb", "bab", "abaa", "bb", "ba", "bbb", "abba", "bbab", "aba", "aaa", "b", "aab", "aaab", "abb", "baab", "babb", "bbaa" };

char s1[1000] = "";

cin.getline(s1, 100);

int y = 0, q = 0, o = 0, q1 = 0,x1=0;

string dog1[1000] = {};

for (int x = 0; x != strlen(s1); x++) {

if (s1[x] == ' ')

q++;

}

for (int j = 0; j != q + 1; j++) {

char mas1[1000] = "";

for (int x = x1, l = 0; x < strlen(s1); l++,x++) {

if (s1[x] == ' ') {

x1 = x+1;

break;

}

mas1[l] += s1[x];

}

q1++;

string mas2 = mas1;

sort(mas2.begin(), mas2.end());

do

{

if (o > 0)

y++;

for (int i = 0; i < mas2.length(); i++) {

for (int x = 0; x < 26; x++) {

if (mas2[i] == eng[x]) {

dog1[y] += mas[x];

o++;

break;

}

}

}

} while (next\_permutation(mas2.begin(), mas2.end()));

}

int t = 0;

for (int i = 0; i != y+1; i++) {

int u = 0;

for (int j = 0; j != y + 1; j++) {

if (dog1[i] == dog1[j]) {

u++;

}

}

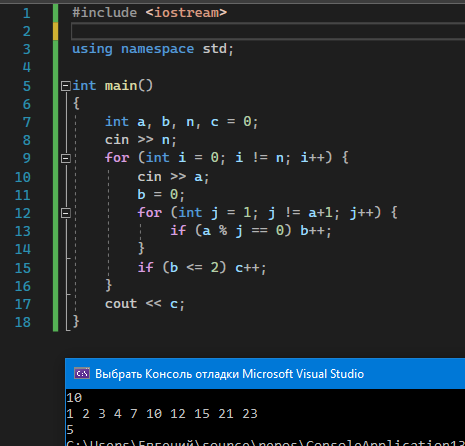
if (u == 1) t++;

}

cout << t;

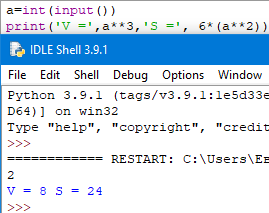
}****

**3)C++**

****

***Python***

***1)Python***

******

***2)Python***

from itertools import permutations

eng=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

mas=['ab','baaa', 'baba', 'baa', 'a', 'aaba', 'bba', 'aaaa', 'aa', 'abbb', 'bab', 'abaa', 'bb', 'ba', 'bbb', 'abba', 'bbab', 'aba', 'aaa', 'b', 'aab', 'aaab', 'abb', 'baab', 'babb', 'bbaa']

s1=str(input())

y=0

q=0

x1=0

q1=0

for x in range (0,len(s1)):

if s1[x]==' ':

q+=1

for j in range (0,q+1):

for x in range (x1,len(s1)):

if s1[x]==' ':

x1=x+1

break

if x>x1:

mas1+=s1[x]

else:

mas1=s1[x]

q1+=1

mas2=sorted(mas1)

r=0

for x in mas2:

if r==0:

mas3=x

else:

mas3+=x

r+=1

if q1==1:

dog1 = [''.join(p) for p in permutations(mas3)]

else:

dog1 += [''.join(p) for p in permutations(mas3)]# в дог1 записал все перестановки букв

print(dog1)

for i in range(len(dog1)):

w = ''

for j in range(len(dog1[i])):

for k in range(26):

if dog1[i][j] == eng[k]:

w+= mas[k]

dog1[i] = w

w = ''

print(dog1)

t=0

for i in range(len(dog1)):#сделано(нахожу уникальные+вывод)

u=0

for j in range(len(dog1)):

if dog1[i]==dog1[j]:

u+=1

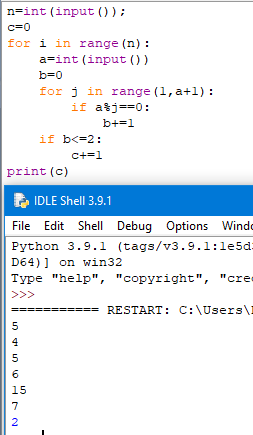
if u==1:

t+=1

print(t)

****

**3)Python**

****

***Haskell***

***1)Haskell***

mygcd (a) = a\*a\*a

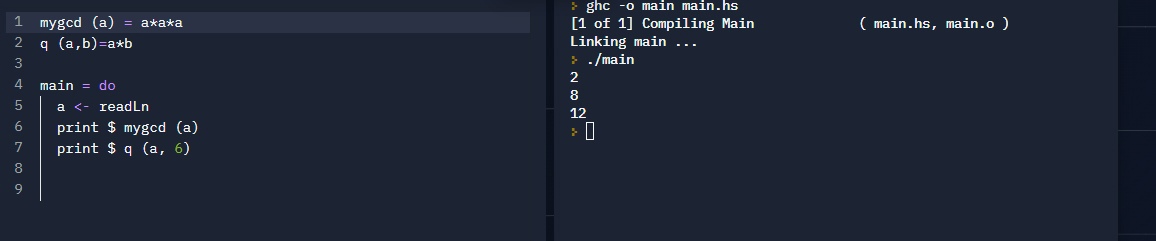
q (a,b)=a\*a\*b

main = do

a <- readLn

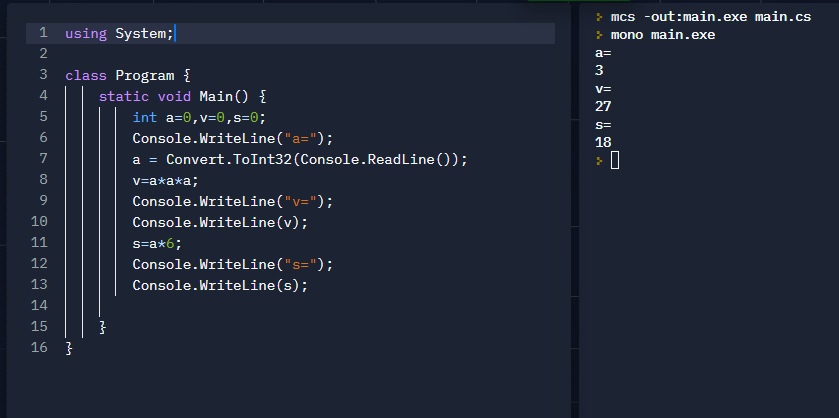
print $ mygcd (a)

print $ q (a,6)



**C#**

**1)C#**



**2)C#**

**using System;**

**using System.Collections.Concurrent;**

**using System.Diagnostics;**

**using System.Threading;**

**using System.Collections.Generic;**

**namespace MiniCube**

**{**

**class GFG**

**{**

**static List<string> list = new();**

**private static void permute(string str,**

**int l, int r)**

**{**

**if (l == r)**

**{**

**list.Add(str);**

**}**

**else**

**{**

**for (int i = l; i <= r; i++)**

**{**

**str = swap(str, l, i);**

**permute(str, l + 1, r);**

**str = swap(str, l, i);**

**}**

**}**

**}**

**public static string swap(string a,**

**int i, int j)**

**{**

**char temp;**

**char[] charArray = a.ToCharArray();**

**temp = charArray[i];**

**charArray[i] = charArray[j];**

**charArray[j] = temp;**

**string s = new string(charArray);**

**return s;**

**}**

**public static void Main()**

**{**

**var dict = new Dictionary<char, string>()**

**{**

**['a'] = "ab",**

**['b'] = "baaa",**

**['c'] = "baba",**

**['d'] = "baa",**

**['e'] = "a",**

**['f'] = "aaba",**

**['g'] = "bba",**

**['h'] = "aaaa",**

**['i'] = "aa",**

**['j'] = "abbb",**

**['k'] = "bab",**

**['l'] = "abaa",**

**['m'] = "bb",**

**['n'] = "ba",**

**['o'] = "bbb",**

**['p'] = "abba",**

**['q'] = "bbab",**

**['r'] = "aba",**

**['s'] = "aaa",**

**['t'] = "b",**

**['u'] = "aab",**

**['v'] = "aaab",**

**['w'] = "abb",**

**['x'] = "baab",**

**['y'] = "babb",**

**['z'] = "bbaa"**

**};**

**string[] s1 = {"ab", "cd"};**

**foreach (var item in s1)**

**{**

**string mas2 = item.ToString();**

**int n = mas2.Length;**

**permute(mas2, 0, n - 1);**

**}**

**var morze = new List<string>();**

**foreach (var item in list)**

**{**

**string newstr = "";**

**for (int i = 0; i < item.Length; i++)**

**{**

**if (dict.TryGetValue(item[i], out string result))**

**{**

**newstr += result;**

**}**

**}**

**morze.Add(newstr);**

**Console.WriteLine(newstr);**

**}**

**int t = 0;**

**for (int i = 0; i != morze.Count ; i++)**

**{**

**int u = 0;**

**for (int j = 0; j != morze.Count ; j++)**

**{**

**if (morze[i] == morze[j])**

**{**

**u++;**

**}**

**}**

**if (u == 1)**

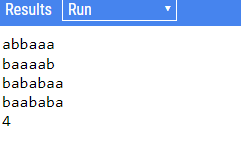
**t++;**

**}**

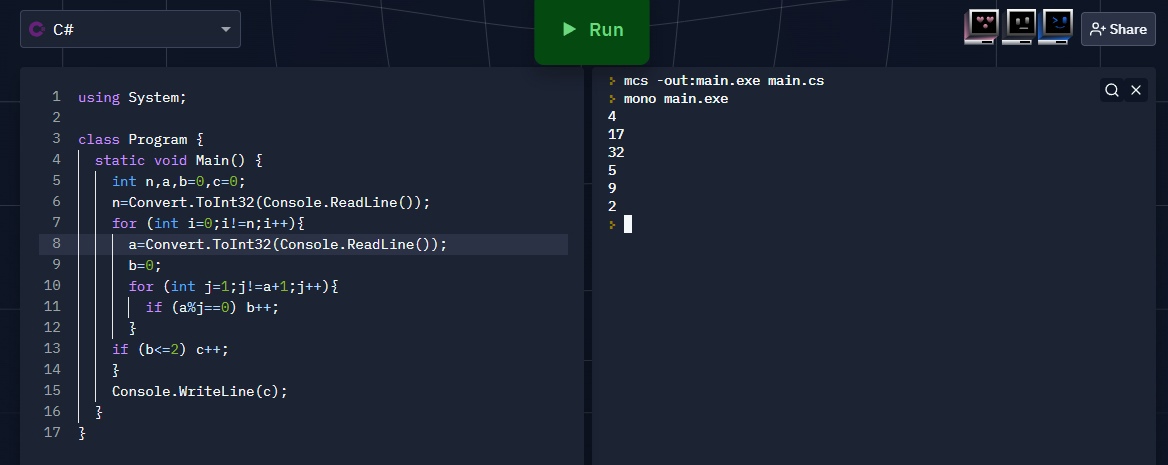
**Console.WriteLine(t);**

**}**

**}**

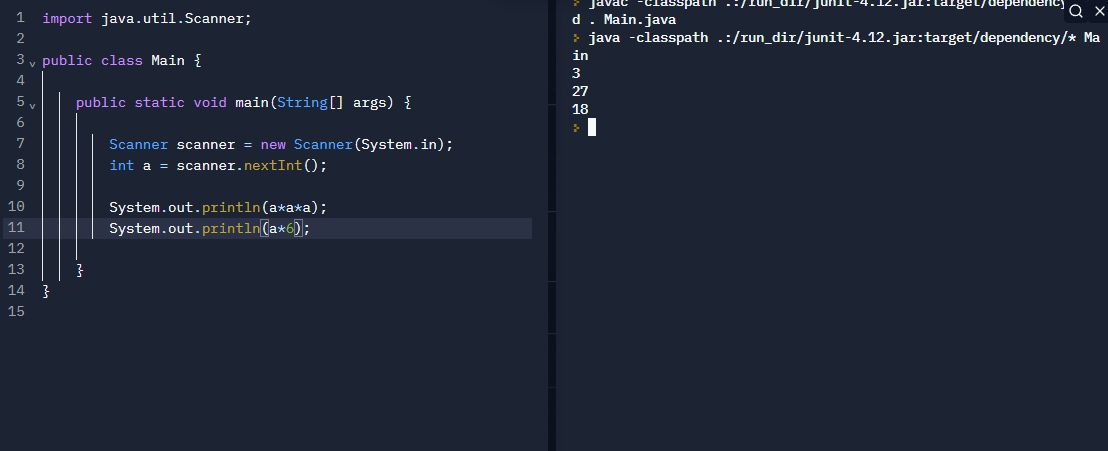
**}**

**3)C#**



**Java**

**1)java**



**2)java**

**public class Main {**

**public static void main(String[] args) {**

**String str = "b";**

**String[] words = str.split(" ");**

**int count = 1;**

**String[] uniques = new String[words.length];**

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

**words[i] = encode(words[i]);**

**}**

**boolean isAppend;**

**uniques[0] = words[0];**

**for (int i = 1; i < words.length; i++) {**

**isAppend = true;**

**for (int j = 0; j < count; j++) {**

**if (isEquals(words[i], uniques[j])) {**

**isAppend = false;**

**break;**

**}**

**}**

**if (isAppend) {**

**uniques[count] = words[i];**

**count++;**

**}**

**}**

**System.out.println(count);**

**}**

**public static String encode(String str) {**

**char[] english = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',**

**'m', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',**

**'y', 'z', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0',**

**',', '.', '?' };**

**String[] morse = { ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..",**

**".---", "-.-", ".-..", "--", "-.", "---", ".---.", "--.-", ".-.",**

**"...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", ".----",**

**"..---", "...--", "....-", ".....", "-....", "--...", "---..", "----.",**

**"-----", "--..--", ".-.-.-", "..--.." };**

**StringBuilder result = new StringBuilder();**

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

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

**if (str.charAt(i) == english[j]) {**

**result.append(morse[j]);**

**result.append(" ");**

**break;**

**}**

**}**

**}**

**return String.valueOf(result);**

**}**

**public static boolean isEquals(String a, String b) {**

**String[] words1 = a.split(" ");**

**String[] words2 = b.split(" ");**

**String[] tmp;**

**boolean result;**

**for (int i = 0; i < words1.length; i++) {**

**for (int j = 0; j < words2.length; j++) {**

**if (words2[j] == "") {**

**continue;**

**}**

**if (words1[i] == "") {**

**break;**

**}**

**tmp = compensate(words1[i], words2[j]);**

**words1[i] = tmp[0];**

**words2[j] = tmp[1];**

**}**

**}**

**for (int i = 0; i < words1.length; i++) {**

**if (words1[i] != "") {**

**return false;**

**}**

**}**

**for (int i = 0; i < words2.length; i++) {**

**if (words2[i] != "") {**

**return false;**

**}**

**}**

**return true;**

**}**

**public static String[] compensate(String a, String b) {**

**String[] result = new String[]{a, b};**

**if (a.equals(b)){**

**result[0] = "";**

**result[1] = "";**

**} else if (a.length() > b.length()) {**

**if (a.startsWith(b)) {**

**result[0] = a.replaceFirst(b, "");**

**result[1] = "";**

**} else if (a.endsWith(b)) {**

**result[0] = a.substring(0, a.length() - b.length());**

**result[1] = "";**

**}**

**} else if (a.length() < b.length()) {**

**if (b.startsWith(a)) {**

**result[0] = "";**

**result[1] = b.replaceFirst(a, "");**

**} else if (b.endsWith(a)) {**

**result[0] = "";**

**result[1] = b.substring(0, b.length() - a.length());**

**}**

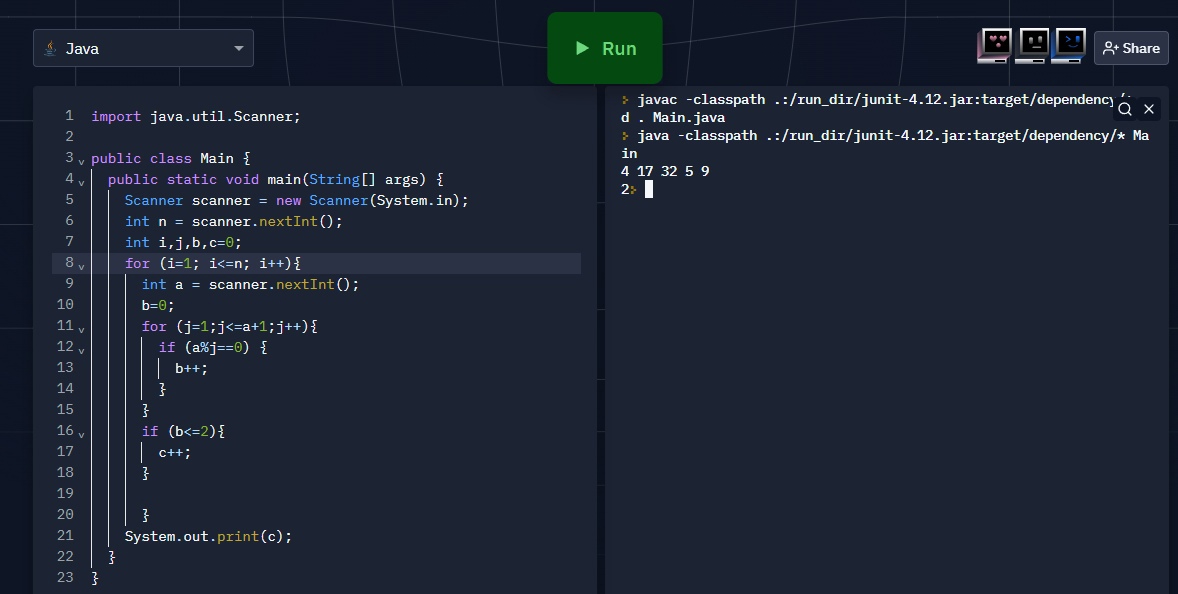
**}**

**return result;**

**}**

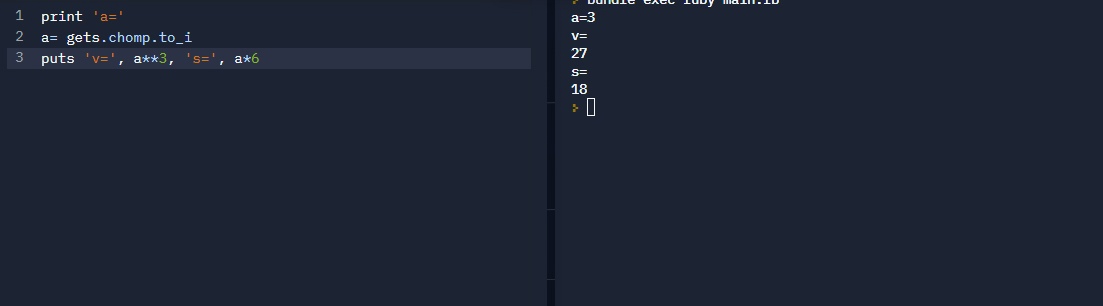
**}  
Input b**

****

**3)**java

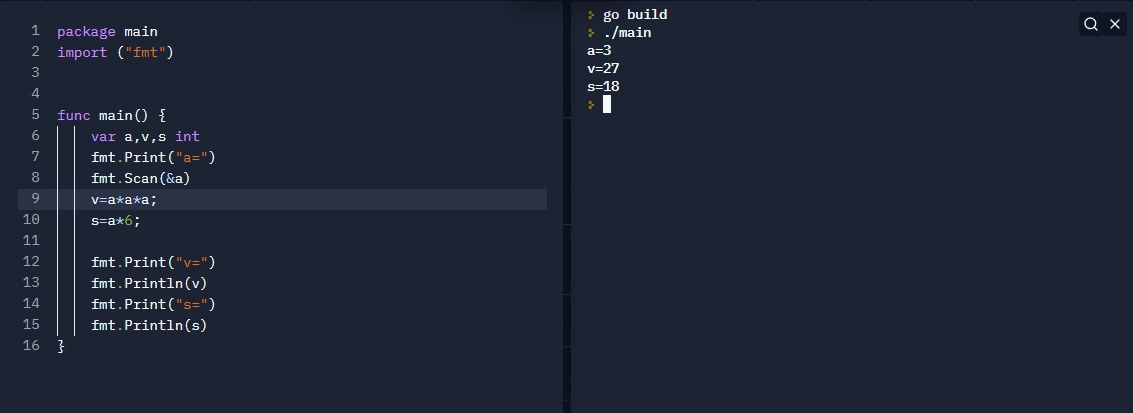
***Ruby***

1)Ruby



***GO***

1)Go



2)Go

package main

import "fmt"

func Perm(a []rune, f func([]rune)) {

perm(a, f, 0)

}

//

func perm(a []rune, f func([]rune), i int) {

if i > len(a) {

f(a)

return

}

perm(a, f, i+1)

for j := i + 1; j < len(a); j++ {

a[i], a[j] = a[j], a[i]

perm(a, f, i+1)

a[i], a[j] = a[j], a[i]

}

}

func main() {

s1:= []string{"ign","ezn","igg","smg"};

eng:= [26]string{"a","b","c","d","e","f","g","h","i","j","k","l","m",

"n","o","p","q","r","s","t","u","v","w","x","y","z"};

mas:=[26]string{"ab","baaa", "baba", "baa", "a", "aaba", "bba", "aaaa", "aa", "abbb", "bab", "abaa", "bb", "ba", "bbb", "abba",

"bbab", "aba", "aaa", "b", "aab", "aaab", "abb", "baab", "babb", "bbaa" };

s2:=[1000]string{""};

s3:=[1000]string{""};

dog1:=[1000]string{""};

y:=0;

j:=0;

for i:=0;i!=len(s1);i++{

Perm([]rune(s1[i]), func(a []rune) {

s2[i]=string(a);

s3[j]=s2[i];

s4:=s3[j];

for r:=0;r!=len(s4);r++ {

for x:=0;x<26;x++{

if string(s4[r])==eng[x]{

dog1[y]+=mas[x];

}

}

}

fmt.Println(dog1[y]);

y+=1;

j+=1;

})

}

t:=0;

for i:=0;i!=y;i++{

u:=0;

for j:=0;j!=y;j++{

if dog1[i]==dog1[j]{

u+=1;

}

}

if u==1{

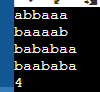
t+=1;

}

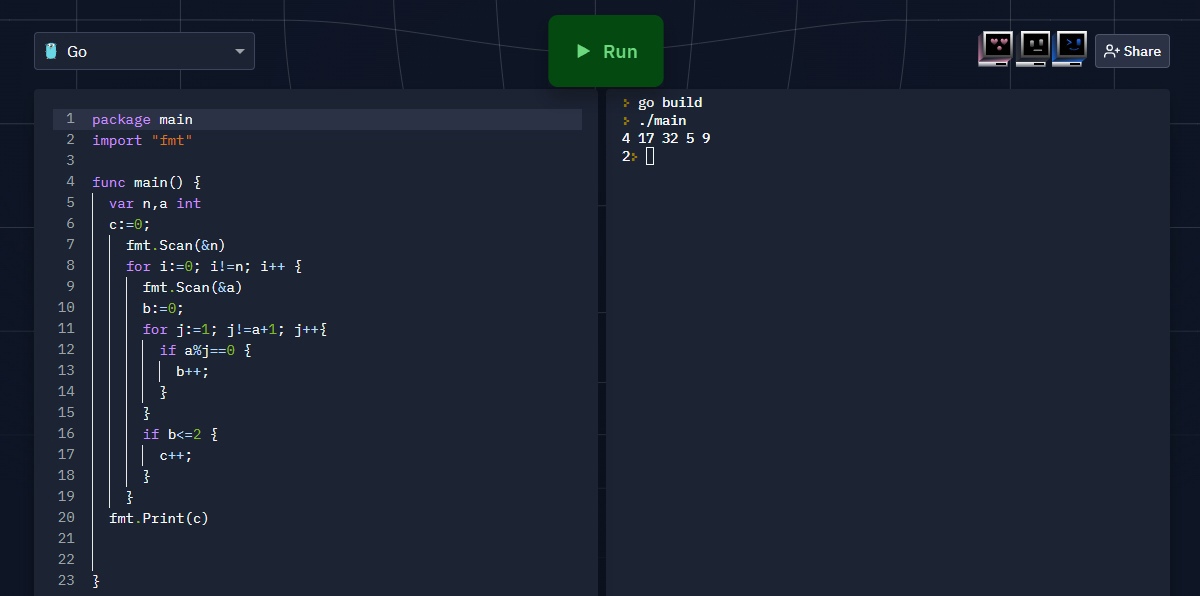
}

fmt.Println(t);

}



3)Go



***Assembler***

include console.inc  
.code

Start: mov ax, 2

Mov ecx, 6

Mov eax, 2  
 mov ebx, eax

Mul ebx

Mul ecx

Outint ax

Outstrln ‘ ‘

Mov ax, 2

Mov eax, 2

Mov ebx, eax

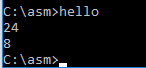
Mul ebx

Mul ebx

Outint ax

Exit

End Start



**Дополнительные задания:**

**1. Топ 10 странных языков программирования:**

1. Brainfuck
2. Chicken
3. Malbolge
4. ArnoldC
5. Ook!
6. Rockstar
7. Shakespeare
8. Piet
9. LOLCODE
10. Glass

**2. Хронология создания ЯП:**

****

**Вывод:** Проделав данную работу, я ознакомился с несколькими видами ввода и вывода данных на нескольких ЯП.