Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №6

по дисциплине «Основы алгоритмизации и

программирования»

Тема: «Подпрограммы»

Выполнил:

студент гр. ПРИ-120

Грачев Д.А.

Приняла:

Шамышева О.Н.

Владимир 2020 г.

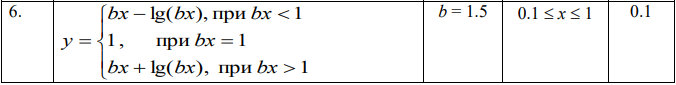
ЦЕЛЬ РАБОТЫ

Освоить разработку модульных программ с использованием процедур и функций на языках Python и C++.

ЗАДАЧА 1(ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Протабулировать функцию y на заданном диапазоне изменения аргумента с заданным шагом.



Листинг программы:

import math

def tabul():

b = 1.5

x = .1

while .1<=x<=1:

if b \* x < 1:

y = round((b\*x - math.log10(b\*x)),3)

elif b \* x > 1:

y = round((b\*x + math.log10(b\*x)),3)

else:

y = 1

print ('x = {} y = {}'.format(x,y))

x = round((x+.1),1)

tabul()

Скриншот работы программы представлен на Рис. 1.

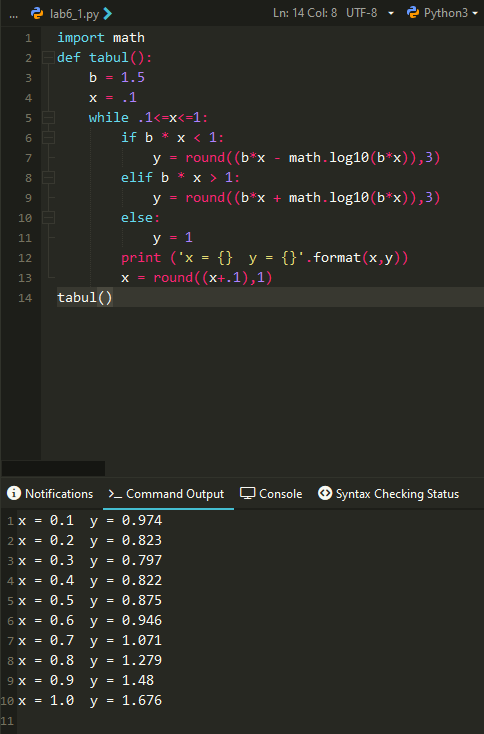
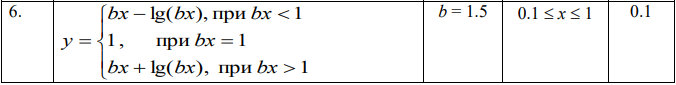


Рисунок 1. Скриншот работы программы.

ЗАДАЧА 1(ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Протабулировать функцию y на заданном диапазоне изменения аргумента с заданным шагом.



Листинг программы:

#include <iostream>

#include <cmath>

using namespace std;

void tabul() {

double b = 1.5, x = 0.1, y;

while ((x >= 0.1) and (x <= 1.1)) {

if (b \* x < 1) {

y = (b \* x - log10(b \* x));

}

else if (b \* x > 1) {

y = (b \* x + log10(b \* x));

}

else {

y = 1;

}

cout << "x = " << x << " y = " << round(y \* 1000) / 1000 << endl;

x = x + 0.1;

}

}

int main() {

tabul();

}

Скриншот работы программы представлен на Рис. 2.

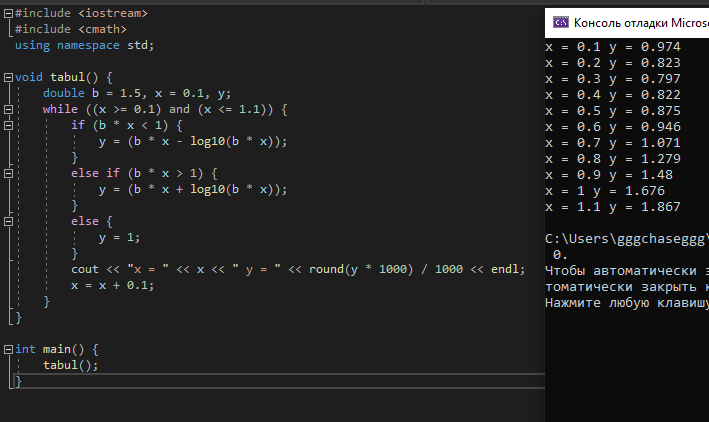


Рисунок 2. Скриншот работы программы

ЗАДАЧА 2(ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Даны два вектора A (a1, a2, …, aN) и B (b1, b2, …, bN). Найти вектор C (c1, c2, …, cN), компоненты которого равны сумме соответствующих компонентов векторов A и B. Найти сумму неотрицательных компонентов вектора C. Печатать вектор C и вычисленную сумму.

Листинг программы:

from random import randint

def suma(n):

a = []; b = []; c= [];suma = 0

for i in range(n):

a.append(randint(-50,50))

b.append(randint(-50,50))

c.append(a[i]+b[i])

if c[i] >= 0:

suma += c[i]

return c,suma

s = 0

n = int(input("Введите длину массивов: "))

c,s = suma(n)

print ("Вектор С: {}\nСумма неотрицательных чисел: {}".format(c,s))

Скриншот работы программы представлен на Рис. 3.

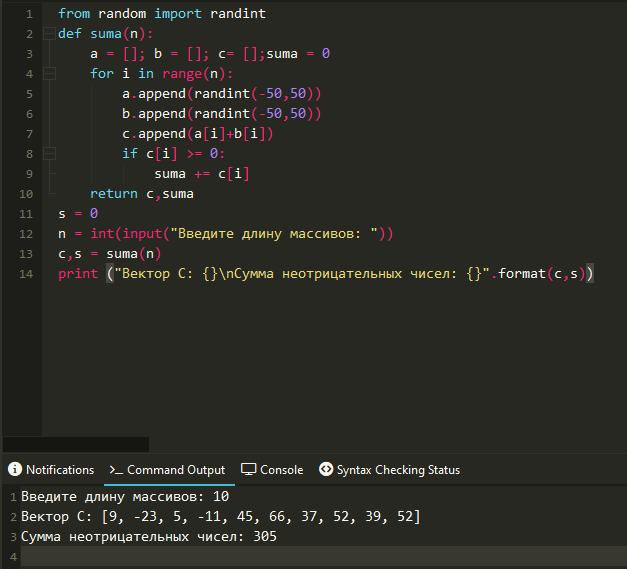


Рисунок 3. Скриншот работы программы.

ЗАДАЧА 2(ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Даны два вектора A (a1, a2, …, aN) и B (b1, b2, …, bN). Найти вектор C (c1, c2, …, cN), компоненты которого равны сумме соответствующих компонентов векторов A и B. Найти сумму неотрицательных компонентов вектора C. Печатать вектор C и вычисленную сумму.

Листинг программы:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int\* c;

int s;

int\* suma(int n) {

int a[10], b[10], c[10], s = 0;

srand(time(NULL));

cout << "Array: ";

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

a[i] = rand() % 101 - 50;

b[i] = rand() % 101 - 50;

c[i] = a[i] + b[i];

cout << c[i] << " ";

if (c[i] >= 0) {

s = s + c[i];

}

}

return c;

}

int n;

int main() {

s = 0;

cin >> n;

c = suma(n);

cout << "\nSum:" << s;

return 0;

}

Скриншот работы программы представлен на Рис. 4.

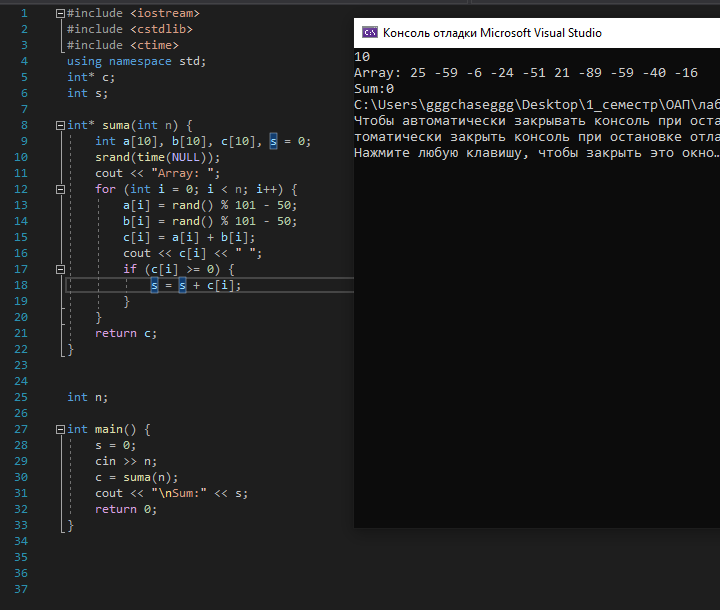


Рисунок 4. Скриншот работы программы

ЗАДАЧА 3(ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Дан двумерный массив K(L, N). Определить номер строки этого массива с максимальным по модулю произведением отрицательных элементов.

Листинг программы:

from random import randint

def zapolnenie(strings\_array,columns\_array):

matrix = [0] \* strings\_array

for i in range(strings\_array):

matrix[i] = [0] \* columns\_array

for j in range(columns\_array):

matrix[i][j] = randint(-50,50)

print (matrix[i][j], end = '\t')

print ('\n')

return matrix

def max\_stroka(array):

max\_v\_str = 0

for i in range(len(array)):

proizvedenie = 1

for j in range(len(array[0])):

if array[i][j] < 0:

proizvedenie \*= array[i][j]

if abs(proizvedenie) > max\_v\_str:

max\_v\_str = abs(proizvedenie)

index\_max\_string = i

return index\_max\_string

strings\_array = int(input("Введите кол-во строк матрицы: "))

columns\_array = int(input("Введите кол-во столбцов матрицы: "))

array = zapolnenie(strings\_array,columns\_array)

index = max\_stroka(array)

print ("Индекс строки с маскимальным по модулю произведением элементов: ", index)

Скриншот работы программы представлен на Рис. 5.

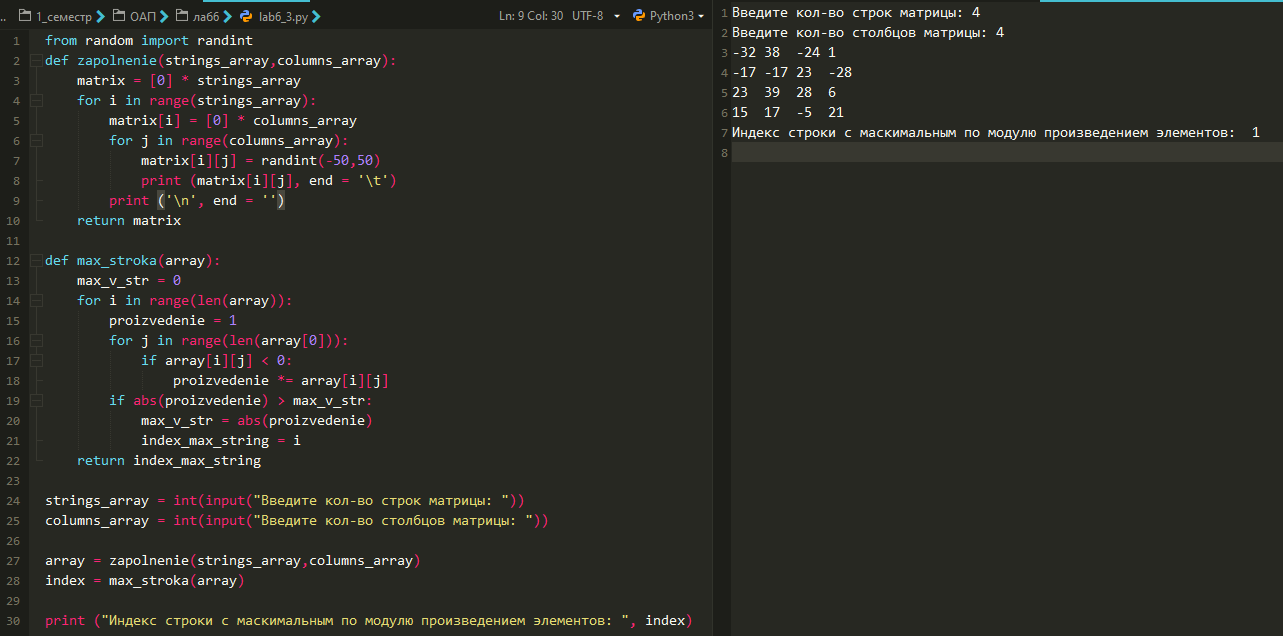


Рисунок 5. Скриншот работы программы.

ЗАДАЧА 3(ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Дан двумерный массив K(L, N). Определить номер строки этого массива с максимальным по модулю произведением отрицательных элементов.

Листинг программы:

#include <iostream>

#include <cmath>

#include <ctime>

#include <cstdlib>

using namespace std;

int\*\* zapolnenie(int count\_row, int count\_columns) {

srand(time(NULL));

int\*\* matrix = new int\* [count\_row];

for (int i = 0; i < count\_columns; i++) {

matrix[i] = new int[count\_columns];

}

for (int row = 0; row < count\_row; row++) {

for (int columns = 0; columns < count\_columns; columns++) {

matrix[row][columns] = rand() % 101 - 50;

cout << matrix[row][columns] << '\t';

}

cout << endl;

}

return matrix;

}

int max\_stroka(int\*\* matrix) {

int max\_v\_str = 0, index\_max\_str;

for (int row = 0; row < sizeof(matrix); row++) {

int proizvedenie = 1;

for (int columns = 0; columns < sizeof(matrix[0]); columns++) {

if (matrix[row][columns] < 0) {

proizvedenie = proizvedenie \* matrix[row][columns];

}

}

if (abs(proizvedenie) > max\_v\_str) {

max\_v\_str = abs(proizvedenie);

index\_max\_str = row;

}

}

return index\_max\_str;

}

int main() {

int count\_row, count\_columns, index;

int\*\* matrix;

cout << "Number of lines: ";

cin >> count\_row;

cout << "Number of columns: ";

cin >> count\_columns;

cout << endl;

matrix = zapolnenie(count\_row, count\_columns);

index = max\_stroka(matrix);

cout << "Index: " << index;

}

Скриншот работы программы представлен на Рис. 6.

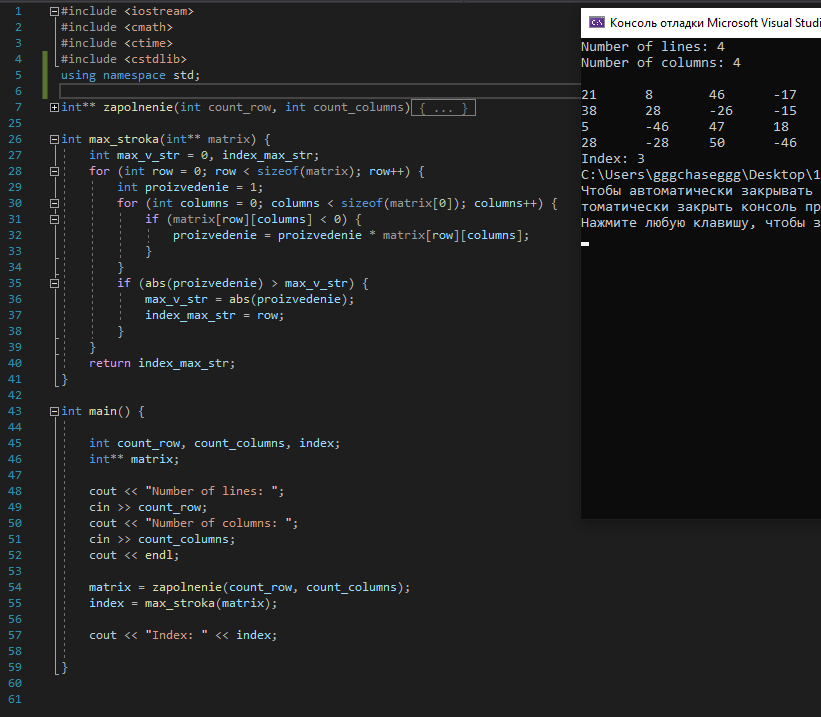


Рисунок 6. Скриншот работы программы

ЗАДАЧА 4 (Вариант №6) на Python

На языке Python выполнить задание:

Найти отношение K-го числа Фибоначчи к N-му.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Системный анализ
   1. Цель разработки

Разработка выполняется в среде Python 3 c учебной целью для получения практических навыков создания подпрограмм.

* 1. Назначение программы

Программа находит отношения K-го числа ряда Фибоначчи к N-му.

* 1. Методы решения

При решении используются следующие функции:

* Вручную созданная функция element()
* Сложение (+)
* Умножение (\*)
* Деление (/)
* Округление(round())
  1. Основные требования к выполняемым функциям

Вывести отношение чисел в формате: «Отношение: число(округленное до 3 знаков после запятой)»

* 1. Набор тестов

Набор тестов к программе представлен на таблице 1.

Таблица 1. Набор тестов

|  |  |  |
| --- | --- | --- |
| Номер теста | Ввод | Вывод |
| 1 | K = 5  N = 7 | Отношение k/n: 0.385 |
| 2 | K = 6  N = 7 | Отношение k/n: 0.615 |
| 3 | K = 7  N = 7 | Отношение k/n: 1.0 |

1. Структурное проектирование
   1. Структура данных

Данные:

* k,n – входные данные типа int
* s – выходные данные типа float
  1. Структура программы

Структура программы в виде блок-схемы представлена на Рис.7



Рисунок 7. Блок-схема структуры программы.

* 1. Используемые подпрограммы

Стандартные процедуры ввода-вывода.

1. Программирование

Листинг программы:

def element(x):

a = [0,1]

for i in range(2,x+1):

a.append(a[i-2]+a[i-1])

return a[x]

k = int(input("Введите k-тое число: "))

n = int(input("Введите n-тое число: "))

s = element(k)/element(n)

print ("Отношение k/n:", round(s,3))

1. Тестирование и отладка

В процессе отладки были устранены синтаксические ошибки. Тестирование по тестам п. 1.4. дало ожидаемые результаты.

Скриншот работы программы представлен на Рис.8.

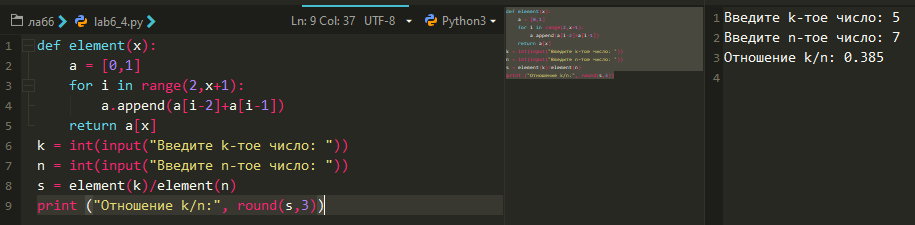


Рисунок 8.Скриншот работы программы.

ЗАДАЧА 4(ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Найти отношение K-го числа Фибоначчи к N-му.

Листинг программы:

#include <iostream>

#include <ctime>

#include <cmath>

using namespace std;

float element(int x) {

srand(time(NULL));

int a[100] = { 0,1 };

for (int i = 2; i < x+1; i++) {

a[i] = a[i-2]+a[i-1];

}

return a[x];

}

int k, n;

float s;

int main()

{

cin >> k;

cin >> n;

s = element(k) / element(n);

cout << round(s\*1000)/1000;

}

Скриншот работы программы представлен на Рис. 9.

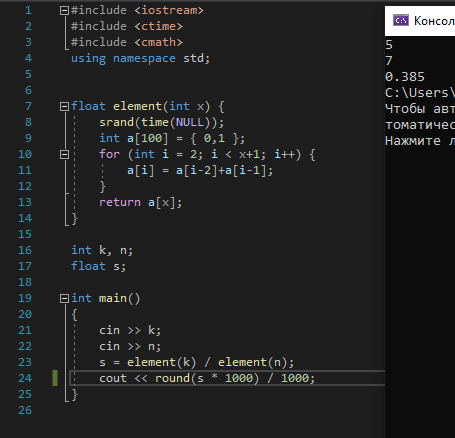


Рисунок 9. Скриншот работы программы

ВЫВОД

В ходе выполнения работы были получены практические навыки по созданию и использованию подпрограмм на языках Python и C++.