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 №4

по дисциплине «Основы алгоритмизации и

программирования»

Тема: «Обработка одномерных массивов»

Выполнил:

студент гр. ПРИ-120

Грачев Д.А.

Приняла:

Шамышева О.Н.

Владимир 2020 г.

ЦЕЛЬ РАБОТЫ

Разработка алгоритма обработки одномерных массивов данных, его программирование на языках С++ и Python.

ЗАДАЧА 6 (ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Дан массив Х (х1, х2, …, хN). Найти и напечатать минимальный элемент массива Х и его номер. Если минимальным элементом оказался последний элемент массива, т переменной v присвоить значение, равное1, в противном случае положить v = 0. Печатать v.

Листинг программы:

import random as rand

n = 10 #Присваивание значения

a = [] #Создание массива

for i in range(n): #Условие цикла

a.append(rand.randint(-50,50)) #Заполнение массива

Min = min(a) #Нахождение минимума

Ind = a.index(Min) #Нахлождения индекса минимума

if Ind == n-1: #Проверка условий

v = 1 #Присвоение значения

else: #Проверка несоответствия условиям

v = 0 #Присвоение значения

print ('Минимальный элемент: {}\nИндекс минимального элемента: {}\nv = {}'.format(Min,Ind,v)) #Вывод

Скриншот работы программы представлен на Рис. 1.

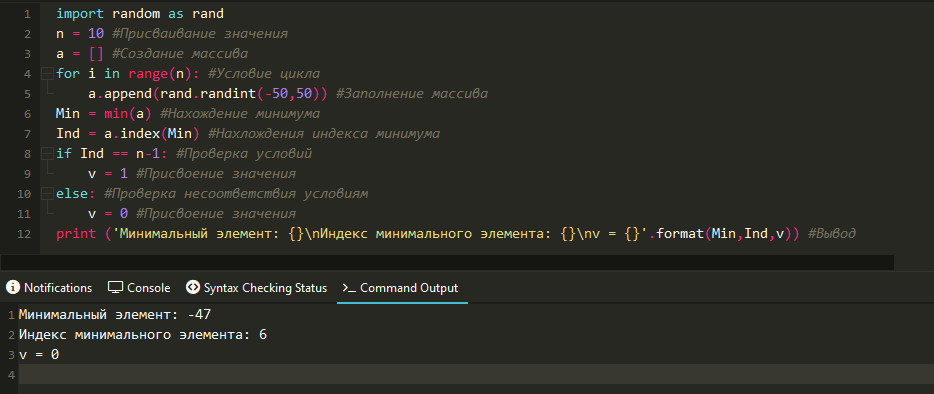


Рисунок 1. Скриншот работы программы.

ЗАДАЧА 1 (ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Дан массив Х (х1, х2, …, хN). Найти и напечатать минимальный элемент массива Х и его номер. Если минимальным элементом оказался последний элемент массива, т переменной v присвоить значение, равное1, в противном случае положить v = 0. Печатать v.

Листинг программы:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int n=10, a[10] , mn = a[0], ind, v; //Объявление переменных

int main() {

srand(time(NULL));

cout << "Array: "; //Вывод массива(начало)

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

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

cout << a[i] << " "; //Вывод массива(конец)

if (a[i] < mn) { //Наождение минимума и его индекса

mn = a[i];

ind = i;

}

}

if (ind == (n-1)){ //Проверка условий

v = 1; //Присвоение значений

}

else {

v = 0; //Присвоение значений

}

cout << "\nMinimum:" << mn << "\nIndex: " << ind << "\nV = " << v; //Вывод

return 0;

}

Скриншот работы программы представлен на Рис. 2.

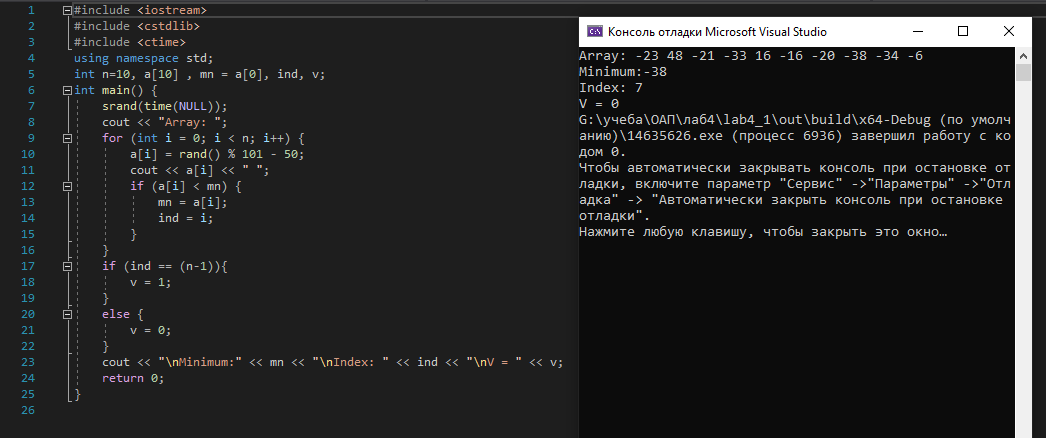


Рисунок 2.Скриншот работы программы.

ЗАДАЧА 2 (Вариант №6)

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.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Системный анализ
   1. Цель разработки

Разработка выполняется в среде Python 3 c учебной целью для получения практических навыков программирования алгоритмов разветвляющейся структуры, обработки арифметических данных.

* 1. Назначение программы

Программа создает новый вектор используя компоненты двух данных.

* 1. Методы решения

При решении используются следующие функции:

* Рандомизация (randint)
* Сложение (+)
  1. Основные требования к выполняемым функциям

Вывод «объяснений» для пользователей в выводе.

* 1. Набор тестов

Набор тестов к программе представлен на таблице 1.

Таблица 1. Набор тестов

|  |  |
| --- | --- |
| Номер теста | Вывод |
| 1 | [-11, -39, -50, 22, 11, -59, 74, 20, 66, 56]  249 |
| 2 | [-8, 15, 18, 67, 4, -39, -8, -35, 15, -59]  119 |
| 3 | [-80, -65, -24, -9, -20, -8, -4, -69, 28, 23]  51 |

1. Структурное проектирование
   1. Структура данных

Данные:

* c,s - выходные действительные числа, тип Int
  1. Структура программы

Структура программы в виде блок-схемы представлена на Рис.3



Рисунок 3. Блок-схема структуры программы.

* 1. Используемые подпрограммы

Стандартные процедуры ввода-вывода.

Библиотека random.

1. Программирование

Листинг программы:

import random as rand

n = 10; a = []; b = []; c = []; s = 0 #Объявление массивов

for i in range(n): #Условие цикла

a.append(rand.randint(-50,50)) #Заполнение массива

b.append(rand.randint(-50,50)) #Заполнение массива

c.append(a[i]+b[i]) #Заполнение массива

if c[i] >= 0: #Проверка условия

s += c[i] #Вычисления

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

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.4.

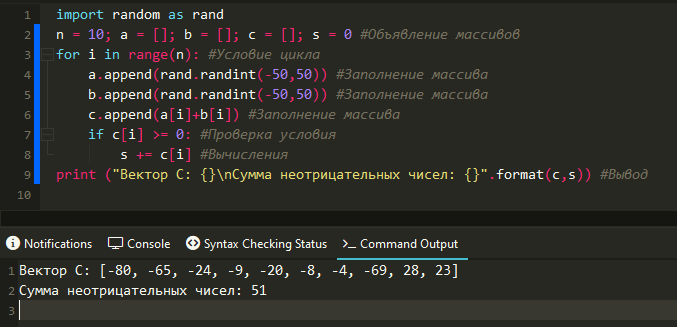


Рисунок 4.Скриншот работы программы.

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 n=10, a[10], b[10],c [10] , s;

int main() {

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];

}

}

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

return 0;

}

Скриншот работы программы представлен на Рис. 5.

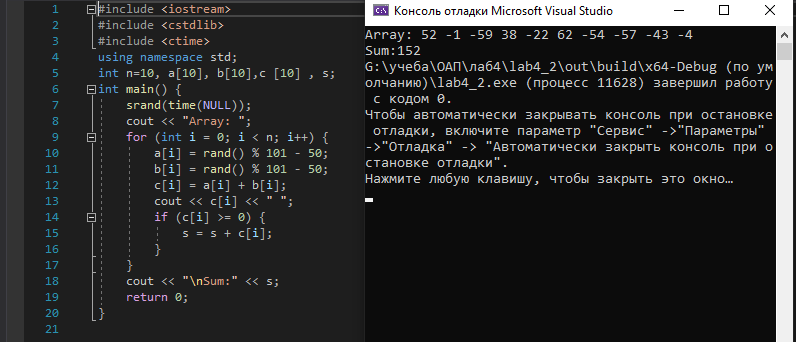


Рисунок 5. Скриншот работы программы.

ЗАДАЧА 3 (ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Дан одномерный массив y (y1, y2, …, y25). Образовать новый массив z (z1, z2, …, z5), элементы которого – суммы пяти последовательных элементов массива y. Печатать z.

Листинг программы:

import random as rand

y = [rand.randint(-50,50) for i in range(25)]

z = []

s = 0

for i in range(5):

for j in range(5):

s += y[i\*5+j]

z.append(s)

s = 0

print (y)

print (z)

Скриншот работы программы представлен на Рис.6

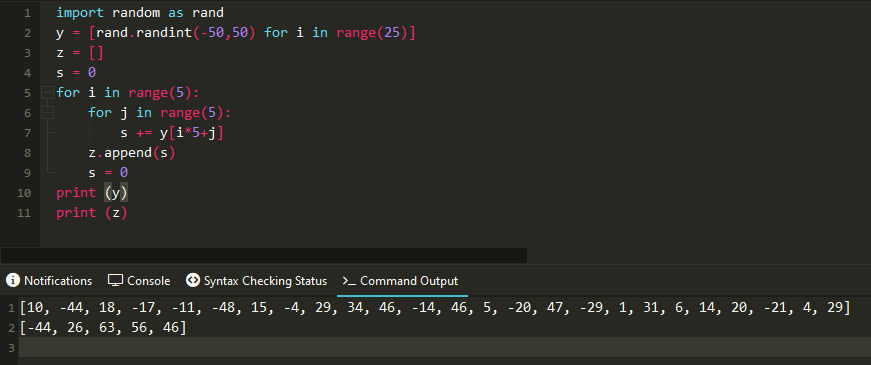


Рисунок 6. Скриншот работы программы.

ЗАДАЧА 3 (ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Дан одномерный массив y (y1, y2, …, y25). Образовать новый массив z (z1, z2, …, z5), элементы которого – суммы пяти последовательных элементов массива y. Печатать z.

Листинг программы:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int y[25], z[5], s = 0;

int main() {

srand(time(NULL));

cout << "Array y: ";

for (int i = 0; i < 25; i++) {

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

cout << y[i] << " ";

}

cout << "\nArray z: ";

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

s = s + y[i \* 5 + j];

}

z[i] = s;

cout << z[i] << " ";

}

return 0;

}

Скриншот работы программы представлен на Рис.7

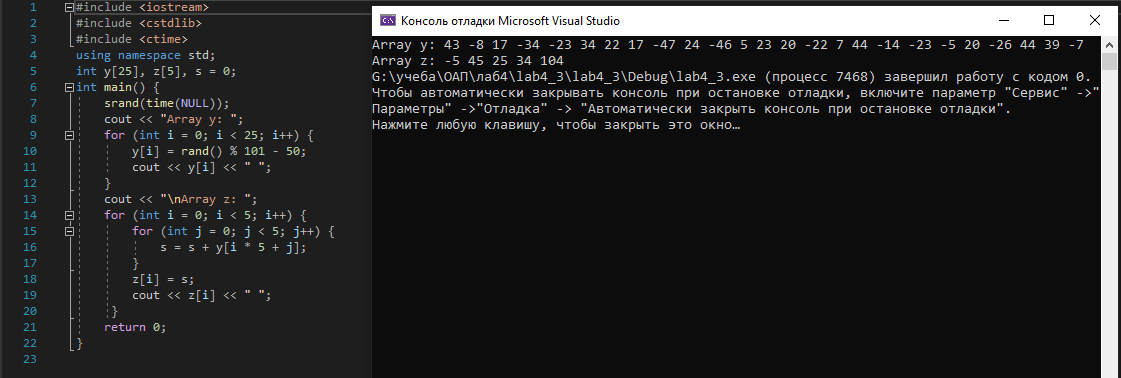


Рисунок 7. Скриншот работы программы.

ЗАДАЧА 4 (ВАРИАНТ №6) на Python

На языке Python выполнить задание:

Увеличить длину массива, вставив заданное число b перед и после минимального положительного элемента массива A[N].

Листинг программы:

import random as rand

b = rand.randint(-50,50)

n = 10

a = []

mn = 51

for i in range(n):

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

if a[i] >= 0 and a[i] < mn:

mn = a[i]

ind = i

print("Изначальный массив:",a)

a.append(0); a.append(0)

for i in range(n-ind+1):

if n-i == ind:

a[n-i+1] = a[n-i]

else:

a[(n+1)-i] = a[n-i-1]

a[ind],a[ind+2] = b,b

print ("Заданное число: {}\nИзмененный массив: {}".format(b,a))

Скриншот работы программы представлен на Рис.8.

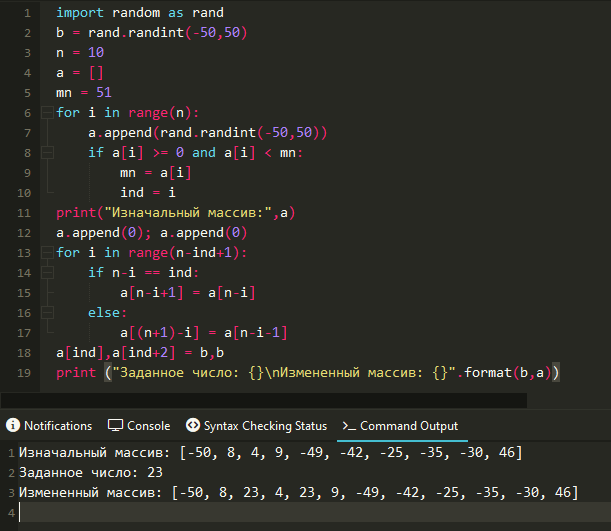


Рисунок 8. Скриншот работы программы.

ЗАДАЧА 4 (ВАРИАНТ №6) на С++

На языке С++ выполнить задание:

Увеличить длину массива, вставив заданное число b перед и после минимального положительного элемента массива A[N].

Листинг программы:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int n = 10, a[12], b = rand() % 101 - 50, mn = 51, ind;

int main() {

srand(time(NULL));

cout << "Array: ";

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

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

if ((a[i] >= 0) and (a[i] < mn)) {

mn = a[i];

ind = i;

}

cout << a[i] << " ";

}

a[10] = 0; a[11] = 0;

for (int i = 0; i < n - ind + 1; i++) {

if (n - i == ind) {

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

}

else {

a[(n + 1) - i] = a[n - i - 1];

}

cout << a[i] << " ";

}

a[ind] = b; a[ind + 2] = b;

cout << a[10] << " " << a[11];

cout << "\nNuber b:" << b;

return 0;

}

Скриншот работы программы представлен на Рис. 9.

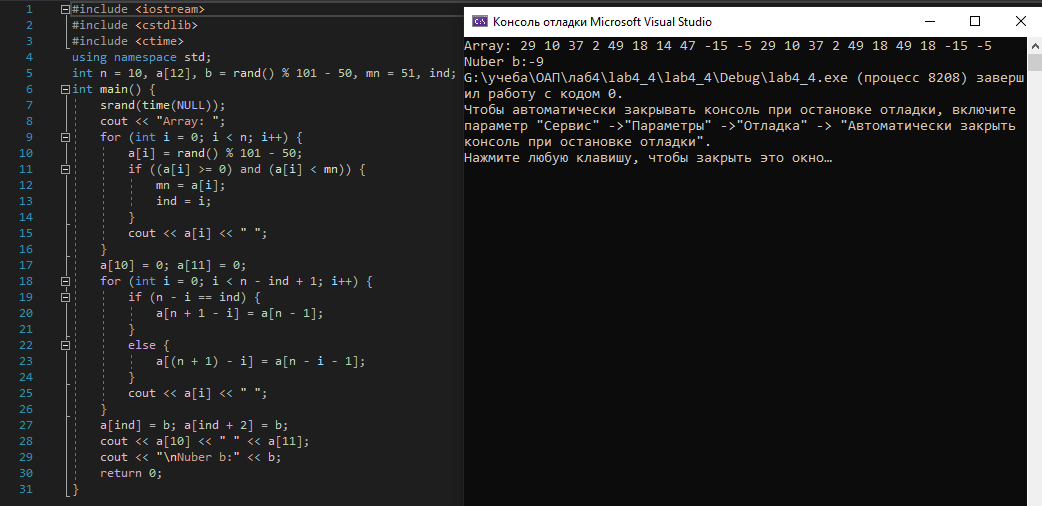


Рисунок 9. Скриншот работы программы.

ВЫВОД К РАБОТЕ

Я научился разрабатывать алгоритмы обработки одномерных массивов данных, программировать их на языках С++ и Python.