Министерство образования РФ

Пермский государственный технический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 5

ПО ИНФОРМАТИКЕ ЗА I СЕМЕСТР

Вариант 9

Выполнил студент:

Главатских Максим Николаевич

Группа РИС-20-1бз

Шифр 20-ЭТФ-659

Кафедра ИТАС:

Полякова Ольга Андреевна

ПЕРМЬ, 2020

**Лабораторная работа №5**

**"** **Функции и массивы"**

Вариант 9.

**Цель:** Организовать обработку массивов с использованием функций, научиться передавать массивы как параметры функций.

**Задача:** Используя функции, решить указанную задачу. Массив должен передаваться в функцию как параметр:

Вариант 9. Написать функцию для вычисления суммы элементов квадратной матрицы, которые расположены ниже главной диагонали. С ее помощью найти максимальное значение такой суммы в n матрицах.

**Текст программы:**

#include <cstdlib>

#include <iostream>

#include <time.h>

#include <math.h>

#define SIZE 5

static int a[SIZE][SIZE];

void CreateArray(int a[SIZE][SIZE]) {

int i, j;

for (i = 0; i < SIZE; i++) {

for (j = 0; j < SIZE; j++) {

a[i][j] = rand() % 100;

}

}

}

void PrintArray(int a[SIZE][SIZE]) {

int i, j;

for (i = 0; i < SIZE; i++) {

for (j = 0; j < SIZE; j++) {

printf("%d ", a[i][j]);

}

printf("\n");

}

}

int MatrixSum1(int m[SIZE][SIZE]) {

int i, j, s = 0;

for (i = 1; i < SIZE; i++) {

for (j = 0; j < i; j++) {

s += m[i][j];

}

}

return s;

}

int main(int argc, char\* argv[]) {

int s = 0, i, t, N;

N = atoi(argv[1]); //количество матриц N вводится в параметре командной строки

srand(time(NULL));

for (i = 1; i <= N; i++) {

CreateArray(a);

PrintArray(a);

t = MatrixSum1(a);

printf("\nSumm:%d\n ", t);

if (t > s) {

s = t;

}

}

printf("\nMaximum %d\n", s);

system("PAUSE");

return EXIT\_SUCCESS;

}

**Результат выполнения программы (N=3):**

0 72 91 65 7

59 23 13 17 77

55 16 74 40 85

18 56 19 89 63

20 87 49 89 35

Summ:468

30 92 77 90 68

40 72 79 7 91

16 46 34 78 0

9 7 57 33 60

5 31 2 71 84

Summ:284

47 22 56 36 57

19 27 1 5 75

25 67 52 35 73

90 0 34 9 78

99 27 48 33 59

Summ:442

Maximum 468