

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Омский государственный технический университет»  
Кафедра «Автоматизированные системы обработки информации и  
управления»

**ОТЧЁТ**  
**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №3**  
по дисциплине «Измерительные средства аналитики программных систем и  
технологий»

Выполнил Ст. гр. ИВТ-234  
Моисеенко Д.В.

Проверил Доцент, к. т. н.  
Зубарев А.А.

Омск 2024

**Задача:** Даны две целые квадратные матрицы шестого порядка. Распечатать элементы главных диагоналей каждой из них и вычислить суммы элементов отдельно главной и побочной диагоналей.

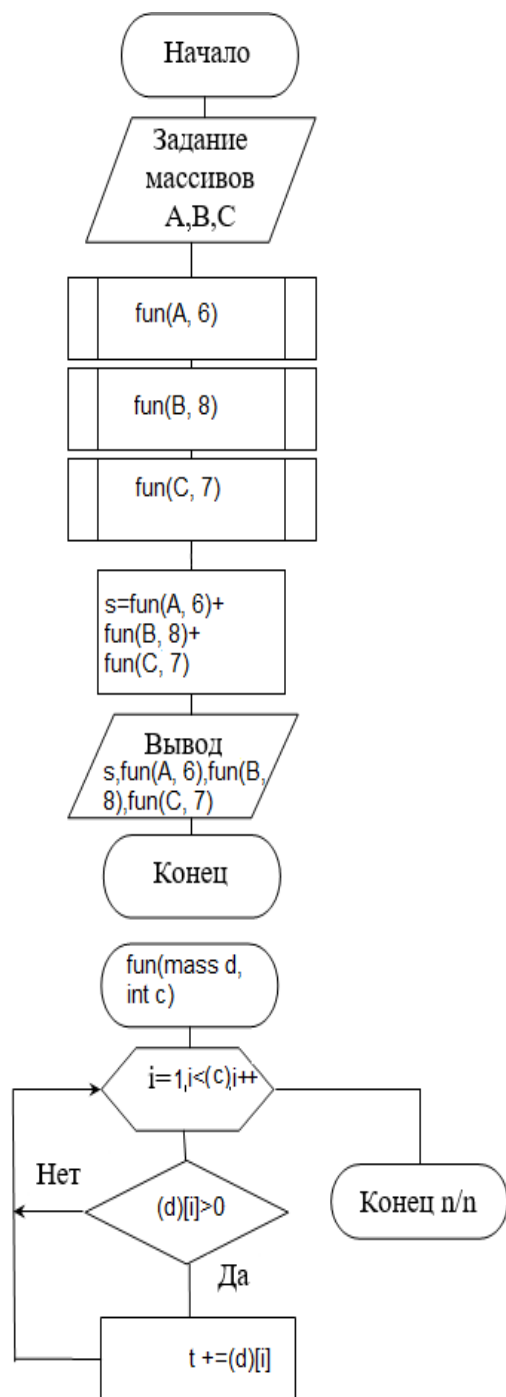
### Текст программы

```
#include <conio.h>
#include <math.h>
#include <stdio.h>
#include <locale.h>
int* f(int c[6][6]) {
    int i, j, z[2];
    z[0] = 0;
    z[1] = 0;
    for (i = 0; i < 6; ++i) {
        for (j = 0; j < 6; ++j) {
            if (i == j) {
                printf(" %d ", c[i][j]);
                z[0] = z[0] + c[i][j];
            }
            if (i + j == 5) z[1] = z[1] + c[i][j];
        }
    }
    printf("\n");
    return z;
}
int main() {
    setlocale(LC_ALL, "RU");
    int a[6][6] = { { 1, 1, 1, 1, 1, 1 }, { 2, 2, 1, 2, 1, 1 }, { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 10 }, { 1, 1, 1, 1, 1, -6 }, { 1, 1, 1, 5, 1, 1 } };

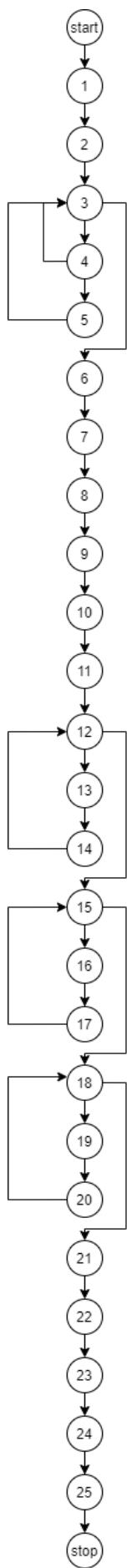
    int b[6][6] = { { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 1 }, { 1, 1, 1, 1, 1, 1 } };

    int* z = f(a);
    printf("%d, %d\n\n", z[0], z[1]);
    int* q = f(b);
    printf("%d, %d", q[0], q[1]);
    return 0;
}
```

### Схема алгоритма



**Граф потока управления**



Номер шага графа	Действие шага
1	<code>int a[6][6] = { {1, 1, 1, 1, 1, 1}, {2, 2, 1, 2, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 10}, {1, 1, 1, 1, 1, -6}, {1, 1, 1, 5, 1, 1} }</code>
2	<code>int b[6][6] = { {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1} }</code>
3	<code>int* z = f(a)</code>
4	<code>int* f(int c[6][6])</code>
5	<code>z[2] = [0, 0]</code>
6	<code>for (i = 0; i &lt; 6; ++i)</code>
7	<code>for (j = 0; j &lt; 6; ++j)</code>
8	<code>if (i == j)</code>
9	<code>printf(" %d ", c[i][j])</code>
10	<code>z[0] = z[0] + c[i][j]</code>
11	<code>if (i + j == 5)</code>
12	<code>z[1] = z[1] + c[i][j]</code>
13	<code>return z</code>
14	<code>printf("%d, %d\n\n", z[0], z[1])</code>
15	<code>int* q = f(b)</code>
16	<code>printf("%d, %d", q[0], q[1])</code>

Формула для вычисления цикломатического числа Мак-Кейба (цикломатической сложности)  $Z(G)$ :

$$Z(G) = e - v + 2p$$

где  $e$  – число дуг ориентированного графа  $G$  (ориентированный граф (орграф) – граф, рёбрам которого присвоено направление. Направленные рёбра называют дугами, рёбрами);

$v$  – число вершин;

$p$  – число компонентов связности графа.

$$Z(G) = 33 - 16 + 4 = 21$$

**Вывод:** Таким образом с помощью формулы для вычисления цикломатического числа Мак-Кейба можно вычислить цикломатическую сложность программы