

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.

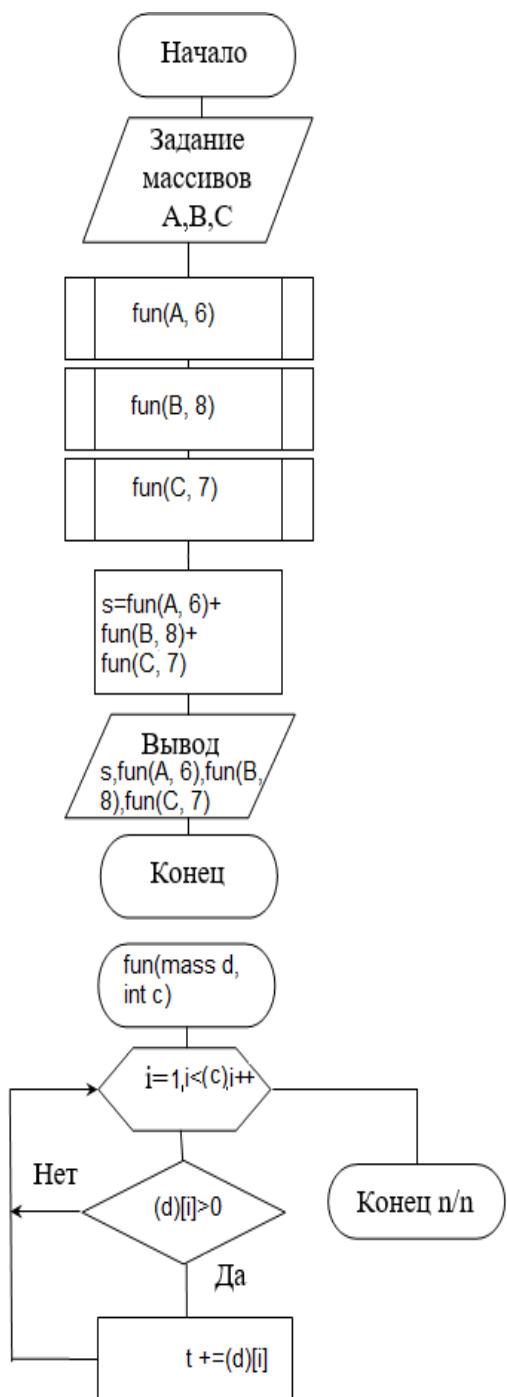
Задача: Даны две целые квадратные матрицы шестого порядка. Распечатать элементы главных диагоналей каждой из них и вычислить суммы элементов отдельно главной и побочной диагоналей.

Текст программы

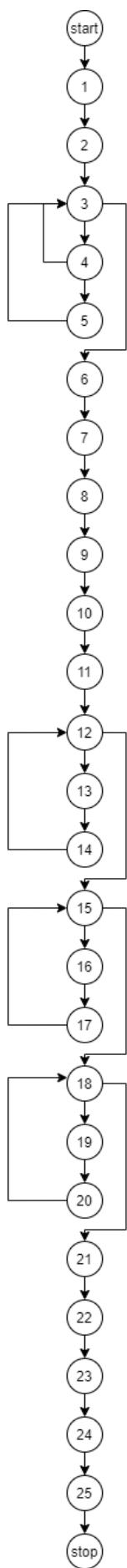
```
#include <conio.h>
#include <math.h>
#include <stdio.h>
#include <locale.h>
int* f(int c[6][6]) {
    int i, j, z[2];
    z[0] = 0;
    z[1] = 0;
    for (i = 0; i < 6; ++i) {
        for (j = 0; j < 6; ++j) {
            if (i == j) {
                printf("%d ", c[i][j]);
                z[0] = 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, -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", 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	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} }
2	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} }
3	int* z = f(a)
4	int* f(int c[6][6])
5	z[2] = [0, 0]
6	for (i = 0; i < 6; ++i)
7	for (j = 0; j < 6; ++j)
8	if (i == j)
9	printf(" %d ", c[i][j])
10	z[0] = z[0] + c[i][j]
11	if (i + j == 5)
12	z[1] = z[1] + c[i][j]
13	return z
14	printf("%d, %d\n\n", z[0], z[1])
15	int* q = f(b)
16	printf("%d, %d", q[0], q[1])

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