

# **Курсова работа по Приложна информатика**

на

Симона Николаева Огнянова

76а група

Фак. No: 471224015

*Специалност: Информатика и софтуерни науки*

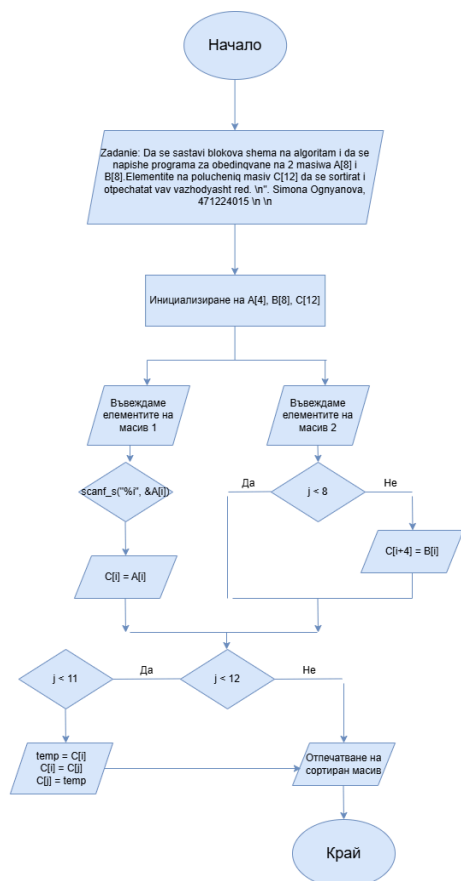
### Задача 1:

Да се състави блокова схема на алгоритъм и да се напише програма за обединяване на два едномерни масива A[4] и B[8]. Елементите на получения масив C[12] да се сортират и отпечатаат във възходящ ред. В началото на програмата да се отпечата заданието, името на студента и фак. номер.

### Алгоритъм:

1. Създаване на нов C проект
2. Чрез printf се отпечатва текстът на заданието, имената и факултетният номер на студента.
3. Декларират се 4 масива - A[4], B[8], C[12]
4. Инициализират се двата масива - A[4] и B[8]. Стойностите се въвеждат от клавиатурата.
5. Създава се for цикъл, който обхожда масива A[4] и добавя всеки елемент в масива C[12].
6. Създава се for цикъл, който обхожда масива B[8] и добавя всеки елемент в масива C[12]
  - Индексирането на C[12] започва от 4 + i, защото първите 4 елемента вече са запълнени с елементите на A[4]
7. Чрез bubble sort се сортира масивът C[12].
8. Чрез for цикъл се отпечатва новополучения масив C[12].

### Блок-схема:



**Решение:**

```
#include <stdio.h>
```

```
void main() {
```

```
    printf("Zadanie: Da se sastavi blokova shema na algoritam i da se napishe programa za  
obedinjvane na 2 masiwa A[8] i B[8]. Elementite na polucheniq masiv C[12] da se sortirat i  
otpechatat vav vazhodyasht red. \n");
```

```
    printf("Simona Ognyanova, 471224015 \n \n");  
    int A[4], B[8], C[12];
```

```
    printf("Enter the elements of the first array: \n");  
    for (int i = 0; i < 4; i++) {  
        scanf_s("%i", &A[i]);  
    }
```

```
    printf("Enter the elements of the second array: \n");  
    for (int i = 0; i < 8; i++) {  
        scanf_s("%i", &B[i]);  
    }
```

```
    for (int i = 0; i < 4; i++) {  
        C[i] = A[i];  
    }
```

```
    for (int i = 0; i < 8; i++) {  
        C[4 + i] = B[i];  
    }
```

```
    int temp;  
    for (int i = 0; i < 12 - 1; ++i)  
    {  
        for (int j = i + 1; j < 12; ++j)  
        {  
            if (C[i] > C[j]) {  
                temp = C[i];  
                C[i] = C[j];  
                C[j] = temp;  
            }  
        }  
    }
```

```
    printf("Sorted array: ");  
    for (int i = 0; i < 12; i++) {  
        printf("%i ", C[i]);  
    }
```

```
}  
}
```

```
Microsoft Visual Studio Debug Console  
Zadanie: Da se sastavi blokova shema na algoritam i da se napishe programa za obedinivane na 2 masiwa A[8] i B[8]. Elementite na poluch  
eniq masiv C[12] da se sortirat i otpechatat vav vazhodyasht red.  
Simona Ognyanova, 471224015  
  
Enter the elements of the first array:  
3  
1  
5  
3  
Enter the elements of the second array:  
12  
3  
22  
55  
4  
3  
12  
32  
Sorted array: 1 3 3 3 3 4 5 12 12 22 32 55  
C:\Users\simon\OneDrive\Desktop\University - practice\Applied-Informatics\CourseProject\x64\Debug\CourseProject.exe (process 13132) exi  
ted with code 0 (0x0).  
Press any key to close this window . . .
```

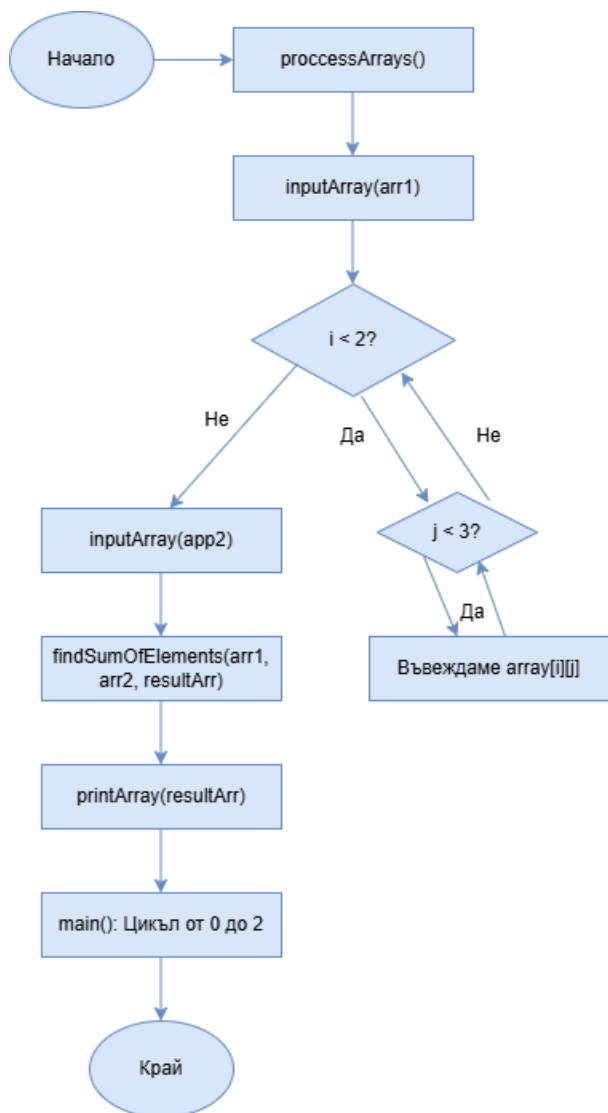
## **Задача. 2**

Да се напише подпрограма функция за сумиране елементите на два двумерни масива. Да се отпечатаат елементите на новополучения масив. Подпрограмата функция да се извика два пъти от главната програма с различни стойности на аргументите.

### **Алгоритъм:**

1. Създава се функция за въвеждане на масив - `inputArray(int array[][3])`.
  - Целта е да се спести написването на един и същи код многократно.
2. Създава се функция `findSumOfElements(int arr1[][3], int arr2[][3], int resultArr[][3])`, която взима двата въведени масива, намира сумата на елементите им и ги добавя в нов масив.
3. Създава се функция, която отпечата новополучения масив - `printArray(int array[][3])`.
4. Създава се функция `processArrays()`, която извиква всички създадени функции.
  - Причината за създаването на функцията е, че в условието се изисква функцията `findSumOfElements(int arr1[][3], int arr2[][3], int resultArr[][3])` да бъде извикана 2 пъти. Чрез този метод се избягва двукратното пренаписване на кода, който се намира в тялото на `processArrays()`.

### **Блок-схема:**



**Решение:**

```
#include <stdio.h>
```

```
void inputArray(int array[][3]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            printf("Element [%i][%i]: ", i, j);
            scanf_s("%i", &array[i][j]);
        }
    }
}
```

```
void findSumOfElements(int arr1[][3], int arr2[][3], int resultArr[][3]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            resultArr[i][j] = arr1[i][j] + arr2[i][j];
        }
    }
}
```

```

    }
}

void printArray(int array[][3]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%i ", array[i][j]);
        }
        printf("\n");
    }
}

void processArrays() {
    int arr1[2][3], arr2[2][3], resultArr[2][3];

    printf("Vavedete elementite na masiv 1: \n");
    inputArray(arr1);

    printf("Vavedete elementite na masiv 2: \n");
    inputArray(arr2);

    findSumOfElements(arr1, arr2, resultArr);
    printf("Novopoluchen masiv: \n");
    printArray(resultArr);
}

void main() {
    for (int i = 0; i < 2; i++) {
        printf("Vavejdane na masivi nomer (%i):\n", i + 1);
        processArrays();
    }
}

```

C:\ Microsoft Visual Studio Debug Console

Vavejdane na masivi nomer (1):

Vavedete elementite na masiv 1:

Element [0][0]: 12

Element [0][1]: 32

Element [0][2]: 43

Element [1][0]: 23

Element [1][1]: 54

Element [1][2]: 65

Vavedete elementite na masiv 2:

Element [0][0]: 45

Element [0][1]: 78

Element [0][2]: 6

Element [1][0]: 45

Element [1][1]: 65

Element [1][2]: 34

Novopoluchen masiv:

57 110 49

68 119 99

Vavejdane na masivi nomer (2):

Vavedete elementite na masiv 1:

Element [0][0]: 23

Element [0][1]: 44

Element [0][2]: 55

Element [1][0]: 66

Element [1][1]: 7

Element [1][2]: 5

Vavedete elementite na masiv 2:

Element [0][0]: 34

Element [0][1]: 12

Element [0][2]: 32

Element [1][0]: 45

Element [1][1]: 66

Element [1][2]: 7

Novopoluchen masiv:

57 56 87

111 73 12