

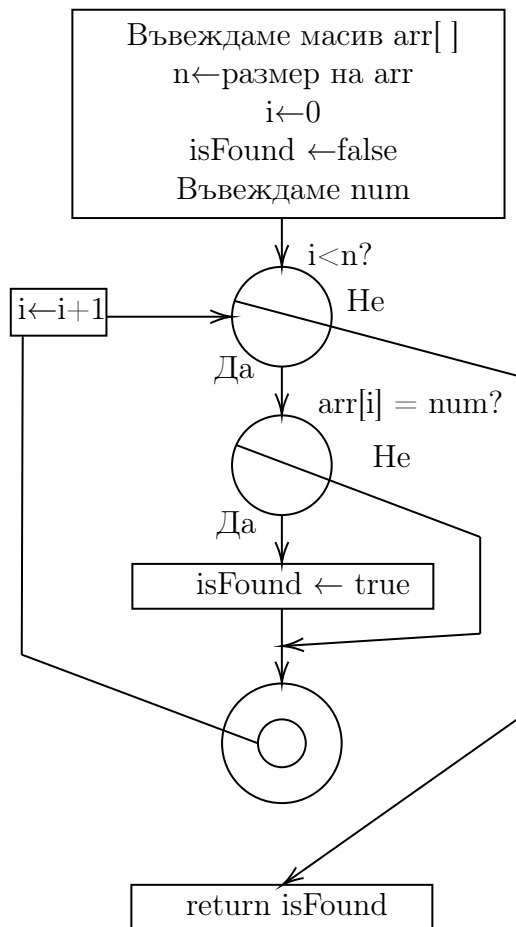
# Увод в алгоритмите и програмирането

## Домашна работа №5

Никола Николов, F106501

**Задача 1:** *Съставете схема на управление и успоредно на нея – програмен текст на претърсване на масив чрез обхождане на целия масив. Изведете броя проверки “ти хикс ли си” на екран. Материалът е подробно даден в лекциите. Пуснете програмата над масив от числа - цифрите на факултетния ви номер (взети като числа), като търсите последователно дали стойностите 1, 2, 3 и 4 се намират вътре или не. Предавате програмен текст, вход и изход – снимка на екран.*

*Решение: [Фигура 1](#)*



```
#include <iostream>

bool simpleSearch(int * arr , int size , int num)

int main()
{
    int arr[] = {1,0,6,5,0,1};
    int size = sizeof(arr)/sizeof(arr[0]);

    std::cout << simpleSearch(arr,size,1) << '\n'
               << simpleSearch(arr,size,2) << '\n'
               << simpleSearch(arr,size,3) << '\n'
               << simpleSearch(arr,size,4) << '\n';

    return 0;
}

bool simpleSearch(int * arr , int size , int num)
{
    bool isFound = false;

    for(int i=0;i<size;i++){
        if(arr[i] == num) isFound = true;
    }

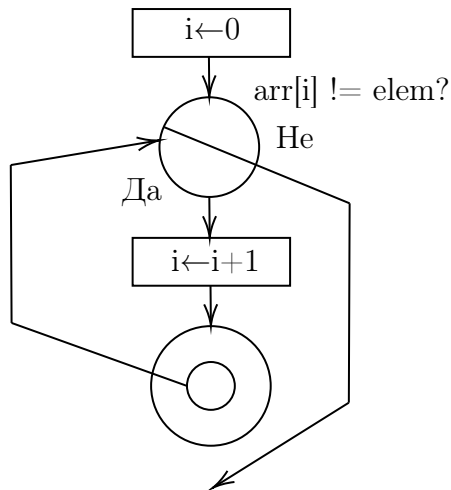
    return isFound;
}
```

Фигура 1: претърсване на масив чрез обхождане на целия масив

```
nn@NNPC:~/Programming/NBU/Algoritmi/Week5/HW5$ ./zad1
1
0
0
0
```

**Задача 2:** Съставете схема на управление и успоредно на нея – програмен текст на търсене в масив дотогава, докато си намери търсената стойност, при използване на „котва“. Материалът е даден подробно в лекциите. Сами съобразете какъв е цикълът и какво трябва да се направи, за да се разбере, след излизане от цикъла, дали търсената стойност е намерена или не. Изведете броя проверки “ти хикс ли си” на екран. Пуснете програмата над масив от букви – буквите от фамилното ви име, като търсите последователно дали стойностите а, б, в и г се намират вътре или не.

Решение:



```

#include <iostream>
#include <cstring>

bool anchorSearch(char * arr,int size, char elem);

int main()
{
    char sirname[] = "Nikolov";
    int size = sizeof(sirname)/sizeof(sirname[0]);
    int arrSize = size + 1;

    char *arr = new char[arrSize];
    strcpy(arr,sirname);

    std::cout << anchorSearch(arr,arrSize,'a') << '\n'
    << anchorSearch(arr,arrSize,'b') << '\n'
    << anchorSearch(arr,arrSize,'v') << '\n'
    << anchorSearch(arr,arrSize,'g') << '\n';

    delete [] arr;
    return 0;
}

bool anchorSearch(char * arr,int size, char elem)
{
    arr[size-2] = elem;
    arr[size-1] = '\0';

    int i=0;
    while(arr[i]!=elem){
        i++;
    }

    std::cout << "Iteration counter for element '"
        << elem <<"' = " << i+1 << '\n';
    if(i==size-2) return false;
    else return true;
}
  
```

Фигура 2: Търсене тип „котва“

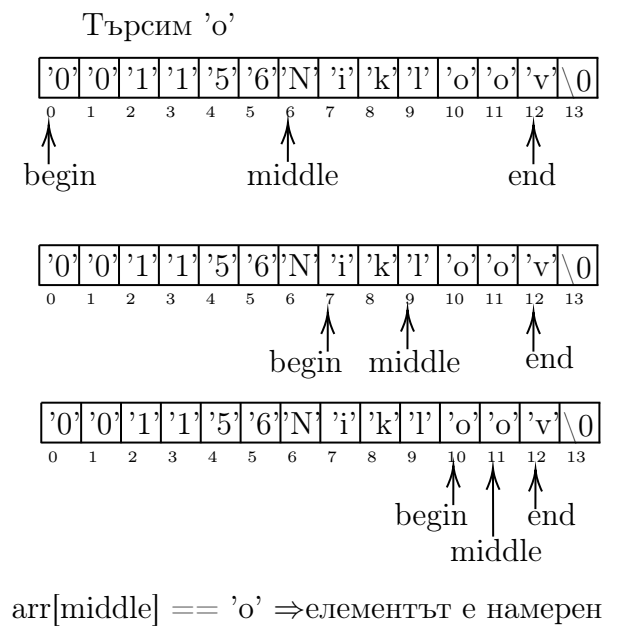
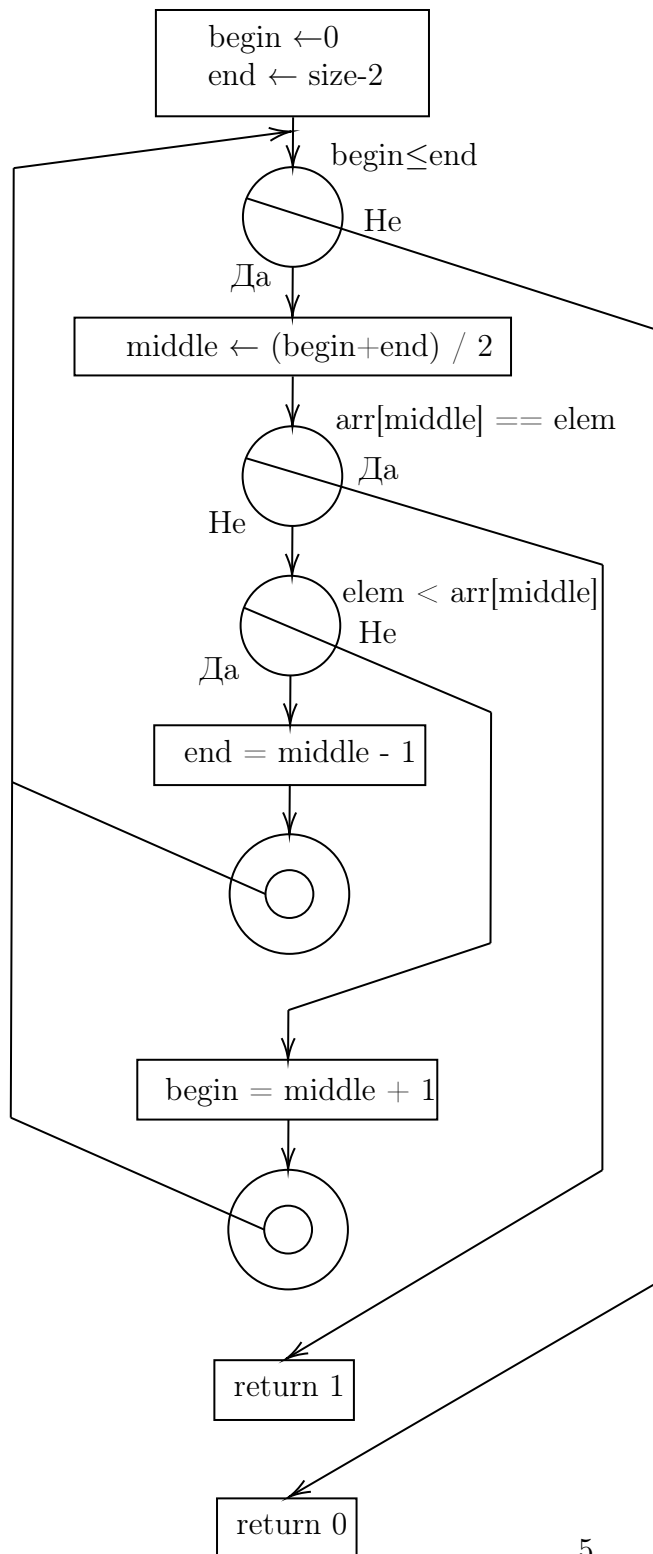
Изход от кода на [Фигура 2](#) :

```
nn@NNPC:~/Programming/NBU/Algoritmi/Week5/HW5$ ./zad2
Iteration counter for element 'a'= 8
0
Iteration counter for element 'b'= 8
0
Iteration counter for element 'v'= 7
1
Iteration counter for element 'g'= 8
0
```

**Задача 3:** *Съставете: А. Опорна схема (това значи схема на масив, означени променливи за индекси за начало и край на претърсваната част, среда, местене на индексите - със стрелки); Б. схема на управление която да съответства на опорната схема и успоредно на нея – програмен текст на дихотомично претърсване на масив. Материалът е даден подробно в лекциите. Изведете на екран как се менят стойностите на променливите съхраняващи индексите за начало и край на подмасива, в който търсенето продължава И на броя проверки “ти хикс ли си”. Пуснете програмата над масив от букви и цифри образуван така – факултетният ви номер и фамилното ви име, залепени, като търсите последователно дали стойностите 9, 8, 7 и я, ю, ъ и о се намират вътре или не. Предавате програмен текст, вход и изход – снимка на екран.*

*Решение:* [Фигура 3](#)

Фигура 3: Дихотомично претърсване на масив.



```

#include <iostream>
#include <algorithm>
#include <cstring>

bool diho(char * arr , int size , char element)
{
    int begin = 0;
    int end = size-2;
    int middle;

    int cnt=1;
    while(begin<=end){
        middle = (begin+end)/2;
        std::cout << "On iteration " << cnt << ": begin= " << begin
                    << " , middle = " << middle << " , end = " << end
                    << '\n';
        if(arr[middle] == element) return 1;
        else if(element < arr[middle]){
            end = middle-1;
        }
        else{
            begin = middle+1;
        }
        cnt++;
    }

    return 0;
}

int main()
{
    char arr[] = "106501Nikolov";
    int size = strlen(arr)+1;
    std::sort(arr, arr + strlen(arr));

    std::cout << "Searching for '9' in " << arr << ":\n" << diho(arr,size,'9') << '\n'
    << "Searching for '8' in " << arr << ":\n" << diho(arr,size,'8') << '\n';
    << "Searching for '7' in " << arr << ":\n" << diho(arr,size,'7') << '\n';
    << "Searching for 'q' in " << arr << ":\n" << diho(arr,size,'q') << '\n';
    << "Searching for 'u' in " << arr << ":\n" << diho(arr,size,'u') << '\n';
    << "Searching for 'y' in " << arr << ":\n" << diho(arr,size,'y') << '\n';
    << "Searching for 'o' in " << arr << ":\n" << diho(arr,size,'o') << '\n';

    return 0;
}

```

Фигура 4: Изход от кода на задача 3.

```
Searching for '7' in 001156Nikloov:
On iteration 1: begin= 0 , middle = 6 , end = 12
On iteration 2: begin= 0 , middle = 2 , end = 5
On iteration 3: begin= 3 , middle = 4 , end = 5
On iteration 4: begin= 5 , middle = 5 , end = 5
0
Searching for 'q' in 001156Nikloov:
On iteration 1: begin= 0 , middle = 6 , end = 12
On iteration 2: begin= 7 , middle = 9 , end = 12
On iteration 3: begin= 10 , middle = 11 , end = 12
On iteration 4: begin= 12 , middle = 12 , end = 12
0
Searching for 'u' in 001156Nikloov:
On iteration 1: begin= 0 , middle = 6 , end = 12
On iteration 2: begin= 7 , middle = 9 , end = 12
On iteration 3: begin= 10 , middle = 11 , end = 12
On iteration 4: begin= 12 , middle = 12 , end = 12
0
Searching for 'y' in 001156Nikloov:
On iteration 1: begin= 0 , middle = 6 , end = 12
On iteration 2: begin= 7 , middle = 9 , end = 12
On iteration 3: begin= 10 , middle = 11 , end = 12
On iteration 4: begin= 12 , middle = 12 , end = 12
0
Searching for 'o' in 001156Nikloov:
On iteration 1: begin= 0 , middle = 6 , end = 12
On iteration 2: begin= 7 , middle = 9 , end = 12
On iteration 3: begin= 10 , middle = 11 , end = 12
1
nn@NNPC:~/Programming/NBU/Algoritmi/Week5/HW5$ █
```