

### Assignment 3

1. Write a program to implement Binary Search.

Ans// C program to implement recursive Binary Search

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

```
// A recursive binary search function. It returns  
// location of x in given array arr[l..r] is present,  
// otherwise -1
```

```
int binarySearch(int arr[], int l, int r, int x)  
{
```

```
    if (r >= l) {
```

```
        int mid = l + (r - l) / 2;
```

```
        // If the element is present at the middle
```

```
        // itself
```

```
        if (arr[mid] == x)
```

```
            return mid;
```

```
        // If element is smaller than mid, then
```

```
        // it can only be present in left subarray
```

```
        if (arr[mid] > x)
```

```
            return binarySearch(arr, l, mid - 1, x);
```

```
        // Else the element can only be present
```

```
        // in right subarray
```

```
        return binarySearch(arr, mid + 1, r, x);
```

```
    }
```

```

        // We reach here when element is not
        // present in array

        return -1;
    }

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };

    int n = sizeof(arr) / sizeof(arr[0]);

    int x = 10;

    int result = binarySearch(arr, 0, n - 1, x);

    (result == -1)

        ? printf("Element is not present in array")

        : printf("Element is present at index %d", result);

    return 0;
}.

```

2. Write a program to implement Sequential Search.

Ans.

```

#include <stdio.h>
#include <conio.h>
main()
{
    int arr[]={12,23,78,98,67,56,45,19,65,9},key,i,flag=0;
    clrscr();
    printf("\nENTER A NUMBER: ");
    scanf("%d",&key);
    for(i=0;i<10;i++)
    {

```

```

        if(key==arr[i])
            flag=1;
    }
    if(flag==1)
        printf("\nTHE NUMBER %d EXISTS IN THE ARRAY",key);
    else
        printf("\nTHE NUMBER %d DOES NOT EXIST IN THE ARRAY",key);
    getch();
}

```

3. Write a program to compare binary and sequential searches. Also, calculate the time taken by each algorithm to find the same number in milliseconds.

Ans.

```

def search(nums, target):
    for i in range(len(nums)):
        if nums[i] == target:
            return i
    return -1

```

```

if __name__ == '__main__':
    nums = [2, 12, 15, 11, 7, 19, 45]
    target = 7
    print(search(nums, target))

```

4. Write a Program to find the maximum element in an array using Linear search.

Ans. // C program to find maximum in arr[] of size n

```

#include <stdio.h>

```

```

// C function to find maximum in arr[] of size n

```

```

int largest(int arr[], int n)
{

```

```
int i;
```

```
// Initialize maximum element
```

```
int max = arr[0];
```

```
// Traverse array elements from second and
```

```
// compare every element with current max
```

```
for (i = 1; i < n; i++)
```

```
    if (arr[i] > max)
```

```
        max = arr[i];
```

```
return max;
```

```
}
```

```
int main()
```

```
{
```

```
    int arr[] = {10, 324, 45, 90, 9808};
```

```
    int n = sizeof(arr)/sizeof(arr[0]);
```

```
    printf("Largest in given array is %d", largest(arr, n));
```

```
    return 0;
}
```

5. Write a Program to find the minimum element in an array using Binary Search.

Ans. #include<stdio.h>

```
int main(){
    int a[50], n, i, key, flag = 0, low, mid, high;
    printf("enter the no: of elements:");
    scanf ("%d",&n);
    printf("enter the elements:");
    for(i=0; i<n; i++)
        scanf( "%d", &a[i]);
    printf("enter a key element:");
    scanf ("%d", &key);
    low = 0;
    high = n-1;
    while (low<= high ){
        mid = (low + high) /2;
        if (a[mid] == key){
            flag = 1;
            break;
        }
        else{
            if (a[mid] > key)
                high = mid-1;
            else
                low = mid+1;
        }
    }
    if (flag == 1)
        printf ("search is successful");
}
```

```
else
    printf("search is unsuccessful");
return 0;
}
```

.