DSA ASSIGNMENT-1

TANISHA KARMAKAR
21051950
CSE 37

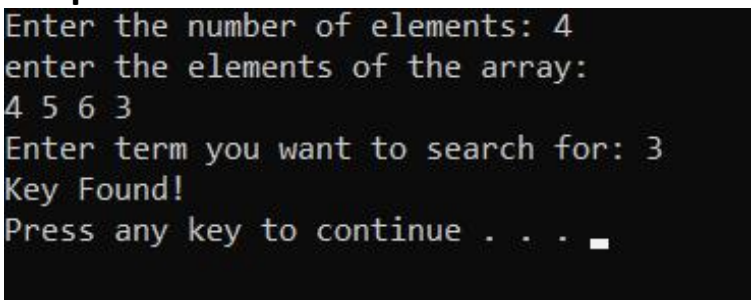
Q1. WAP to create a dynamic array and perform linear search function.

Ans.

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

int search(int a[], int n, int key) {
  int i;
for ( i = 0; i < n; ++i){
    if (a[i] == key){
    return i;
  }
}
    return -1;
}


int main() {
  int n, i, x;
  int *arr;
  printf("Enter the number of elements: ");
  scanf("%d", &n);
   arr=(int*) malloc (sizeof(int)*n);
   printf("enter the elements of the array: \n");
   for(i=0;i<n;i++){
     scanf("%d", &arr[i]);
   }
printf("Enter term you want to search for: ");
```

```
scanf("%d", x);
int pos = search(arr, n, x);
if (pos != 1)
   printf("Key Found! ");
else
   printf("Key not found \n");
return 0;
}
```

Output:



Q2. WAP to static array and write a sort function for bubble sort.

Ans.

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

void swap(int* x, int* y)
{
   int temp = *x;
   *x = *y;
   *y = temp;
}

void sort(int a[],int n){
   int i, j;
   for (i = 0; i < n - 1; i++)
      for (j = 0; j < n - i - 1; j++)
         if (a[j] > a[j + 1])
```

```
            swap(&a[j], &a[j + 1]);
}



int main()
{
   int n;
   printf("Enter elements of array: ");
   scanf("%d", &n);
int arr[n];
   for (int i = 0; i < n; ++i) {
         printf("Enter the element %d  - ",i+1);
         scanf("%d",&arr[i]);
      }

printf("\n");
   sort(arr,n);
printf("Sorted Array - \n");
   for (int i = 0; i < n; i++)
      printf(" %d ",arr[i]);



   return 0;

  }
```

Output:

```
Enter elements of array: 5
Enter the element 1  - 4
Enter the element 2  - 6
Enter the element 3  - 1
Enter the element 4  - 2
Enter the element 5  - 9

Sorted Array -
 1  2  4  6  9
Press any key to continue . . .
```

Q3. WAP to merge two sorted array into one sorted array.
Ans.

```
#include <stdio.h>
int main(){
int arr1[100],arr2[100],m,n;
printf("Enter the size of the 1st array: ");
scanf("%d",&m);
printf("Enter the size of the 2nd array: ");
scanf("%d",&n);
printf("Enter the elements in the 1st array: ");
for (int i = 0; i < m; i++)
{
scanf("%d",&arr1[i]);
}
printf("Enter the elements in the 2nd array: ");
for (int i = 0; i < n; i++)
{
scanf("%d",&arr2[i]);
}
sort(arr1,m);
sort(arr2,n);
merge(arr1,arr2,m,n);

return 0;
}
void merge(int arr1[] , int arr2[],int m,int n){
   int arr3[100];
   for (int i = 0; i < m; i++)
   {
      arr3[i]=arr1[i];
   }
   for (int i = 0; i < n; i++)
   {
      arr3[m+i]=arr2[i];
   }
```

```
    sort(arr3,(m+n));
    printf("Merged Array: \n");
    for (int i = 0; i < m+n; i++)
    {
        printf("%d  ",arr3[i]);
    }

}
void sort(int arr[],int n){
    for (int i = 0; i < n; i++)
    {
        for (int j = i+1; j < n; j++)
        {
            if(arr[i]>arr[j])
            {
                int temp = arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```

Output:

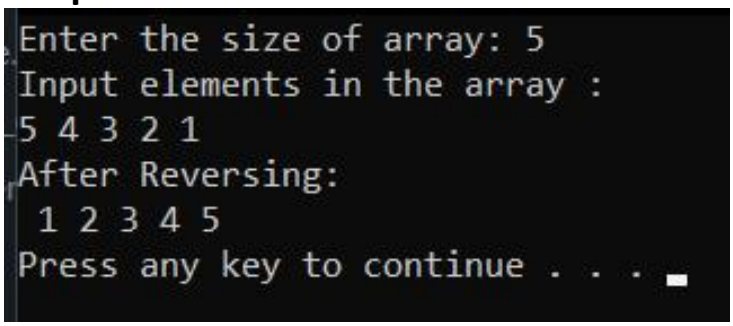
```
Enter the size of the 1st array: 4
Enter the size of the 2nd array: 3
Enter the elements in the 1st array: 2 5 6 1
Enter the elements in the 2nd array: 3 8 7
Merged Array:
1  2  3  5  6  7  8
Press any key to continue . . .
```

Q4. WAP to take an array and reverse the array.

Ans.

```
#include <stdio.h>
int main ()
{
  int arr[100];
    int i, n, temp=0;

    printf("Enter the size of array: ");
    scanf("%d", &n);

    printf("Input elements in the array :\n");
    for (i = 0; i < n; i++) {
       scanf("%d", &arr[i]);
    }

    for ( i = 0; i <= n/2; i++)
    {
      for (int j = n-1-i; j>=0; j--)
      {
        temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;
        break;
      }

    }
    printf("After Reversing:\n ");
    for ( i = 0; i < n; i++)
    {
      printf("%d ",arr[i]);
    }

return 0;
}
```

Output:



Q5. WAP to find the largest element and count the occurrence of the largest element.

Ans.

```
#include <stdio.h>
int main ()
{
  int arr[10];
    int i, n, max,count=0;

    printf("Enter the size of array: ");
    scanf("%d", &n);

    printf("Input %d elements in the array :\n", n);
    for (i = 0; i < n; i++) {
       scanf("%d", &arr[i]);
    }

    max=arr[0];
    for (int i = 0; i < n; i++)
    {
       if(arr[i]>max)
       max=arr[i];
    }
    for (int i = 0; i < n; i++)
    {
       if(max==arr[i])
       count++;
```

```
    }
    printf("The Largest Element: %d",max);
    printf("\nThe Number of times it occured: %d",count);

    return 0;
}
```

Output: