Assignment-6

Ch. Sripathi Harshavardhan
AP19110010410
CSE-F

1.

Program:-

```
#include <stdio.h>
void sort(int a[], int n)
{
  int i, j, temp;
  for(i=0; i<n; i++)
  {
   for (j=0; j<n; j++)
    {
     if (a[i] < a[j])
      {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
      }
    }
  }
}
int binary (int a[], int e, int n)
{
   int i=0, j=n-1, mid;
   while (i<=j)
   {
      mid = (i+j)/2
     if (a[mid] == e)
        return mid+1
     else
     {
       if (e < a[mid])
         j = mid-1;
```

```
        else
            i=mid+1;
        }
    }
    if (i>j)
    {
        return 0;
    }
}
int main()
{
    int n, i, a[20], f, e, m1, m2;
    printf("Enter the no.of elements of array");
    scanf("%d", &n);
    f = binary(a,e,n);
    if (f!=0)
    {
    printf("Enter the elements of array in");
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);
        sort(a,n);
    for (i=0; i<n; i++)
        printf("%d", a[i]);
    printf("Enter the element to find in array");
    scanf("%d", &e);
    f = binary(a,e,n);
    if (f!=0)
    {
    printf("element is found at %d position", f);
    }
```

```
else
{
printf("Element not found\n");
}
Printf("Enter the position of array to find sum
                                    and product\n");
scanf("%d %d", &m1, &m2);
m1--;
m2--;
Printf("The sum is %d", a[m1] + a[m2]);
Printf("The product is %d", a[m1] * a[m2]);
}
```

Output:-

```
Enter the no. of elements of array: 4
Enter the elements of array
2
5
9
8
5 is enter the element to find in array -
Element is found at 1st position.
Enter position of array to find sum & product
1
3
The sum is 8
The product is 15.
```

2)

Program:-

```
#include <stdlib.h>
#include <stdio.h>
#define MAX_SIZE 10
void merge_sort(int,int)
void merge_array(int,int,int,int);
```

```
else
{
printf ("Element not found \n");
}
Printf ("Enter the position of array to find sum
                                    and product\n");
scanf ("%d %d", &m1, &m2);
m1--;
m2--;
Printf ("The sum is %d", a[m1] + a[m2]);
Printf (" The product is %d", a[m2] * a[m2]);
}
```

Output:-

```
Enter the no. of elements of array: 4
Enter the elements of array
2
5
9
8
5 is enter the element to find in array -
Element is found at 1st position.
Enter position of array to find sum & product
1
3
The sum is 8
The product is 15.
```

2)

Program:-

```
#include <stdlib.h>
#include <stdio.h>
#define MAX_SIZE 10
void merge_sort (int, int)
void merge_array (int, int, int, int);
```

Output:-

Simple Merge sort example - functions and array

Enter 5 elements for sorting

8
6
5
3
1

Your Data: 8 6 5 3 1

Sorted Data: 1 3 5 6 8

find the product of kth elements from first and last where k

2

Product = 24

3) Insertion Sort:-

It is a simple sorting algorithm that works the way we sort playing cards in our hands.

Algorithm

// sort an array of size n.

insertion sort (arr, n)

loop from i=1 to n-1

a) Pick element arr[i] and insert it into sorted sequence arr[0 - - - i-1]

Ex:- 12, 10, 13, 4, 8

for above example, for i=1 (second element of array) to 4 (last element of array).

```
if (i<j)
{
 m= (i+j)/2;
 merge_sort (i,m);
 merge_sort (m+1,j);
 merge_array (i,m,m+1,j);
 }
}
 void merge_array (int a, int b, int c, int d)
{
  int t[50];
  int i=a, j=c, k=0;
 while (i<=b && j<=d)
 {
  if (arr_sort[i]<arr_sort[j])
    t[k++] =arr_sort [i++];
   else
    t[k++] =arr_sort [j++];
  }
while (i<=b)
  t[k++] = arr_sort [i++];
 while (j<=d)
   t[k++] =arr_sort [j++];
 for (i=0, j=0; i<=d; i++,j++)
  arr_sort [i] = t[i];
```

i=1, since 10 is smaller than 12, move 12 and insert 10 before 12.

i=2 since 13 will be remain at its position as all elements in array are smaller than 13.
10, 12, 13, 4, 8.

i=3 4 will move to the beginning and other elements from 10 to 13 will move one position ahead of their current position.
4, 10, 12, 13, 8

i=4 8 will move to position after 4, and elements from 10 to 13 will move one position ahead.
4, 8, 10, 12, 13.

Selection sort:

It sorts by an array by repeatedly finding the minimum element from unsorted part and putting it into the beginning. The algorithm maintains two subarrays in given array.

1) The subarray which is already sorted.
2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element from the unsorted subarray is picked and moved to the sorted subarray.

example:
arr[ ] : 40 30 10 20 5
// find the minimum element in arr[0...4]
// and place it in beginning
5 30 10 20 40

// find the minimum element in arr[1---4]
// and place it at beginning of arr[1---4]

5 10 30 20 40

// find the minimum element in arr[2---4]
// and place it at beginning of arr[2---4]

5 10 20 30 40

// find the minimum element in arr[3---4]
// and place iat at beginning of arr[3---4]

5 10 20 30 40.

4.

```
#include <stdio.h>
#include <conio.h>
int main()
{
int arr[50], i, j, n, temp, sum=0, product=1;
printf("Enter total no of elements to store:");
scanf("%d", &n);
printf("Enter %d elements:",n);
for(i=0; i<n; i++)
  scanf("%d", &arr[i])
printf("Sorting array using bubble sort \n");
for(i=0; i<(n-1); i++)
{
for(j=0; j<(n-1-i); j++)
  {
  if (arr[j] > arr[j+1]
    {
    temp = arr[j]
    arr[j] = arr[j+1];
} } arr[j+1] = temp;
```

```
printf("All array elements sorted sucessfully\n");
Printf("Array elements in ascending order:\n\n");
for(i=0;i<n;i++)
 {
  printf("%d\n", arr[i]);
 }
printf("array elements in alternate order\n");
 for(i=0; i<=n; i=i+2)
 {
   printf("%d\n", arr[i]);
 }
 for( i=1; i<=n ; i+2).
 {
   sum = sum + arr[i];
 }
printf("The Sum of odd position elements are %d\n",sum);
 for(i=0; i<=n; i=i+2);
 {
    product *= arr[i];
 }
printf("The product of even position elements are %d\n",
                                                product);

 getch();
 return();
}
```

Output:-

```
Enter total no.of elements to store : 5
 Enter  5 elements : 9
  7
  5
  4
  1
```

Sorting array using bubble sort
All array elements sorted successfully!
Array elements in ascending order.
1
4
5
7
9

Array elements in Alternate order.
9
7
5
4
1

The sum of odd position elements are : 15
The product of even position elements are : 28

5)

```
#include <stdio.h>
#include <stdlib.h>
void Binarysearch(int arr[], int num, int first, int last)
{
int mid;
 if (first > last)
   {printf ("Number is not found");
   }
 else
   {
    mid = (first + last) / 2;
     if (arr[mid] = num)
     {
      printf ("element is found at index %d", mid);
      exit(0);
     }
```

```
    else if (arr[mid] > num)
    {
     Binary search (arr, num, first, mid-1);
    }
    else
     {
      Binary search (arr, num, mid+1, last);
     }
   }
}
void main()
{
 int arr[100], beg, mid, end, i, n, num;
 Printf("Enter size of an array");
 Scanf("%d", &n);
 printf("Enter values in sorted sequence \n");
 for (i=0; i<n; i++)
 {
  Scanf("%d", &arr[i]);
 }
 beg=0;
 end=n-1;
 Printf("Enter a value to be search:");
 Scanf("%d", &num);
 Binary search (arr, num, beg, end);
}
```

Output:

```
Enter the size of an array 4
Enter the values in sorted sequence
1
3
5
7
Enter a value to be search: 7
Element is found at index: 3
```