DSA Assignment

K. Akash
CSE - H
AP19110010755

1)
1)
a)
b)

1)
```
# include <stdio.h>.
void main (int, a[], int n)

{
        int i, j, temp.
        for (i=0 ; i<n; i++)
        {
          for (j = i+1; i<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(j>i)
  {
   return 0;
  }
}
int main()
{
 int n,i,a[20],f,e,m1,m2;
  printf("Enter the no of elements of array");
 scanf("%d",&n);
 printf("Enter the elements of array \n")
  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 elements 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]);

}
```

problem 1.

2) Sort the array using merge sort where elements
2) are taken from the user and find the
product of kth elements from ~~first~~ [illegible] the [illegible]

```
C program for merge sort*|

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

Merger two subarrays of arr[]

First subarray is arr [l .. m]

second subarray is arr [m+1 .. r] void

merge (int arr[], int l, int m, int r)

{
     int i, j, k
       int n1 = m - l + 1;
       int n2 = r - m;

       int L [n1], R [n2]

         for (i=0 ; i<n1; i++)
          l[i] = 0; i<n
          L[i] = arr [l+i];
       for (j=0;  j<n2 ; j++)

          R [j] = arr [m+1 +j];
          i = 0;
          j = 0;
          k = l;
          while (i<n1 && j<n2)
```

```
{ if (L[i] <= R[j])
  {
   arr[k] = L[i];
    i++;
  }
  else
  {
    arr[k] = R[j];
     j++;
   }
    k++;
  }
  while (i<n1)
  {
   arr[k] = L[i];
    i++;
    k++;
  }
  while (j<n2)
  {
   arr[k] = R[j]
    j++;
    k++;
   }
  }
  void merge sort (int arr[], int l, int r)
   {
     if (l<r)
    {
```

```
Same as (l+r)/2, but avoids overflow for
  large l and h
 int m = l+(r-l)/2;
  Sort first and second halves.
  merge sort (arr, l, m);
   merge sort (arr, m+1, r);
   merge (arr, l, m, r);
  }
 }
 void printArray (int A[], int size)
 {
  int i;
   for (i=0; i<size; i++)
    printf ("%d", A[i]);
    printf ("\n");
 }
 int main ()
 {
  int arr[5];
  int i;
  int arr_size = sizeof(arr)/sizeof(arr[0]);
 for (i=0; i<arr_size; i++){
   printf ("Enter the elements");
   Scanf ("%d", &arr[i]);
```

```
}
printf ("Given array is \n");
printArray (arr, arr-size)',
~~merge Sorr~~
mergeSort(arr, 0, arr-size-1);
printf ("\n sorted array is \n");
printArray (arr, arr-size);
int k;
printf(" Enter the value of k");
Scanf ("%d", &k);
int fromfirst = arr [k-1];
int fromlast = arr [5-(k)];
printf("%d", fromlast * fromfirst);
return 0;
}
```

3) Discuss Insertion sort and selection sort with examples.

The selection sort algorithm sorts an array by repeadedly finding the minimum element & from unsorted pair and putting it at the beginning. The algorithm maintains two subarrays in a given array.

The subarray which is already sorted.

Remaining subarray which is unsorted.

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

For example :-

arr [] = 64, 25, 12, 22, 11

Find the minimum element in arry [0..4] and place it at beginning

11, 25, 12, 22, 64.

Find the minimum element in array [1..4] and place it at beginning of arr [1...4]

11, 12, 22, 25 64

Find the minimum element in arr [2 . . 4]

and placot it at beginning of arr [2 . 4]

11 12 22 25 64

Find the minimum element in arr [3 . . 4]

and place it at beginning of arr [3 . . . 4]

11 12 22 25 64

Insertion sort is simple sorting algorithm that works the way we sort plyong cards in our hands.

Algorithm

Sort an arr[] of size n

Insertion sort (arr, n)

Loop from i=1 to n-1

a) pick eldelement arr[i] and insert it into sorted sequence arr [0 . . i-1]

For Example

12, 11, 13, 5, 6

let ~~as~~ us loop for i=1 (to 4

i=1. since 11 is smaller than 12, move 12 and insert 11 before 12

11, 12, 13, 5, 6

i=2. 13 will same remain at its positions as all elements in A[0 . . . 1] are smaller than 13. 12.

11, 12, 13, 5, 6

i=3. 5 will move to the beginning and all other elements from 11 to 13 will move on position a head of their current position.

. 5, 11, 12, 13, 6

i=4. 6 will move to position after 5. and elements from 11 to 13 will move one position ahead of their ~~current~~. current position

5, 6, 11, 12, 13.

4. Sort the arry using bubble sort where elements are taken from the user and display the elements.

i) . in alternate order.

ii) . sum of elements in add positions and product of elements in even positions

iii) elements which are divisible by m

where where m is taken from the user.

Ans
```
# include <stdio.h>
   void main ()
 {
   int a[100], n, i, j, temp, sum =0,
 prod = 1, m;
 printf ("Enter number of elements \n");
 scanf ("%d", &n);
 printf ("Enter %d integers \n", n);
   for (i=0; i<n; i++)
 ~~Scanf~~
   {
    scanf ("%d", &a[i]);
  }
   for (i=0; i<n-1; i++)
  {
   for (j=0; j<n-i-1; j++)
  {
   if (a[j] > a[j+1])
   {
    temp = a[j]
    a[j] = a[j+1];
    a[j+1] = temp;
  }
  }
 }
 printf ("Sorted list in ascending order");
```

```
for (i=0; i<n; i++)
{
printf ("%d\n", a[i]);
}
printf (" the alternate order is ");
for (i=0; i<n; i++)
 { if (i%2 ==0)
 {
  printf ("%d ", a[i]);
 }
}
for (i=0; i<n; i++)
{
 if (i%2 != 0)
 {
  Sum o = sumo + a[i];
 }
}
printf (" sum of odd Index is %d ", sumo)
 for (i=0; i<n; i++)
 {
   if (i%2 ==0)
   {
      prod = prod * a[i];
   }
 }
```

```
printf (" product of odd index is %d ", prod)
printf (" enter the value of m/n");
scanf ("%d ", &m);
 for (i=0 ; i<n ; i++)
 {
   if (a[i]%m ==0)
   {
   printf ("%d ", a[i]);
  }
 }
}
```

5) write a recursive program to implement binary search?

Ans

```
# includ <stdio.h>
 int recursive Binary search (int arr [],
  int start _index ; int end -index , int .element)
 {
 if (end-index >= start_index){
  int middle = start_index + (end -index_
          start_index) / 2 ;
    if (array [middle] == element)
       return middle ;
     if array [middle] > element)
return recursive Binarysearch (array,
start_index, middle-1, element);
return recursive Binarysearch (array, middle+1,
end_index, element);
}
return -1;
}
int main (void)
{
 int array[] = {1,4,7,9,16,56,70};
 int n=7;
 int element=9;
 int found_index =recursive Binarysearch
     (array, 0, n-1, element);
 if (found_index == -1){
 {
  printf ("Element not found in the array");
 }
 else
 {
  printf ("element found at index :
      %d", found_index);
 }
 return 0;
}
```