DSA ASSIGNMENT - 6

G: Mitra Datta
AP19110010552
CSE - H

1.) Take the elements from the user and sort them in descending order and do the following.

a. Using Binary search find the element and the location in array where the element is asked from user.

Ans.

```
#include<stdio.h>
void main()
{
    int n,i,j,temp,first,last,middle,k,b=0;
    printf("Enter number of elements you want to sort:");
    scanf("%d",&n);
    int a[n];
    for(i=0; i<n; i++)
    {
        printf("Enter a[%d]:",i);
        scanf("%d",&a[i]);
    }
    for(i=0; i<n; i++)  // Bubble sort for descending order
    {
        for(j=0; j<n-1; j++)
        {
            if(a[j]<a[j+1])
            {
                temp=a[j+1];
                a[j+1]=a[j];
                a[j]=temp;
            }
        }
    }
```

```
printf("Enter element to search:");
scanf("%d",&k);
first = n-1;        // Binary search for descending Arry
last = 0;
middle = (first + last)/2;
while (last <= first)
{
    if (a[middle] < k)
        first = middle - 1;
    else if (a[middle] == k)
    {
        b = 1;
        break;
    }
    else
        last = middle + 1;
    middle = (first + last)/2;
}
if (b == 1)
    printf("%d found at position %d \n", k,
                                    middle+1);
else
    printf("Element not found");
}
```

Output:

```
Enter number of elements you want to sort: 4
Enter a[0]: 11
Enter a[1]: 3
Enter a[2]: 25
Enter a[3]: 47
Enter element to search: 47
47 found at position 1
```

(After Sorting in descending order)

b.) Ask the user to enter any two locations print the sum and product of values at those locations in the sorted array.

Ans.

```
#include<stdio.h>
void main()
{
    int n,i,j, temp,m,k;
    printf("Enter number of elements to sort:");
    scanf("%d",&n);
    int a[n];
    for(i=0; i<n; i++)
    {
        printf("Enter a[%d]:", i);
        scanf("%d",&a[i]);
    }
    for(i=0; i<n; i++)
    {
        for(j=0; j<n-1; j++)
        {
            if(a[j] < a[j+1])
            {
                temp= a[j+1];
                a[j+1]= a[j];
                a[j]= temp;
            }
        }
    }
printf("Enter two element positions:");
scanf("%d,%d",&m,&k);
if(m>n || k>n)
    printf("Wrong input");
else if(m<0 || n<0)
    printf("Wrong input");
```

```
    else
        printf("Sum, Product are %d,%d",
                a[m-1]+a[k-1], a[m-1]*a[k-1]);
}
```

Output: Enter number of elements to sort: 5

```
Enter a[0]: 12
Enter a[1]: 25
Enter a[2]: 5
Enter a[3]: 37
Enter a[4]: 1
Enter two element positions: 1,2
Sum, Product are: 62, 925 (After sorting in descending order)
```

2.) Sort the array using merge sort where elements are taken from the user and find the product of kth elements from first and last where k is taken from the user.

Ans.

```
#include <stdio.h>
#include <conio.h>
void merge_array(int arry[], int a, int b, int c, int d)
{
    int temp[20];
    int i=a, j=c, k=0;
    while(i<=b && j<=d)
    {
        if(arry[i]<arry[j])
            temp[k++]=arry[i++];
        else
            temp[k++]=arry[j++];
    }
    while(i<=b)
        temp[k++]=arry[i++];
    while(j<=d)
        temp[k++]=arry[j++];
    for(i=a,j=0; i<=d; i++,j++)
        arry[i]=temp[j];
}
void merge_sort(int arry[], int i, int j)
{
    int m;
    if(i<j)
    {
        m=(i+j)/2;
        merge_sort(arry, i, m);
        merge_sort(arry, m+1, j);
        merge_array(arry, i, m, m+1, j);
    }
}
```

```
void main()
{
    int i,n,k;
    printf(" Enter number of elements to sort:");
    scanf("%d", &n);
    int arry[n];
    for(i=0; i<n; i++)
    {
        printf(" Enter arry[%d]:",i);
        scanf(" %d ", &arry[i]);
    }
    for(i=0; i<n; i++)
        printf(" %d", arry[i]);
    merge_sort(arry, 0, n-1);
    printf(" In Sorted Data:");
    for(i=0; i<n; i++)
        printf("%d", arry[i]);
    printf(" Enter k position:");
    scanf("%d", &k);
    if( k>0 && k<n)
        printf(" Product is %d", arry[k-1] *
                                    arry[n-k]);
    else
        printf(" Wrong input");
}
```

Output: Enter number of elements to sort: 3

Enter arry[0]: 37 , Enter a[1]: 5 , Enter a[2]: 9

37 5 9 Sorted Data: 5 9 37

Enter k position: 2

Product is. 81

3.) Discuss Insertion sort and Selection sort with Examples.

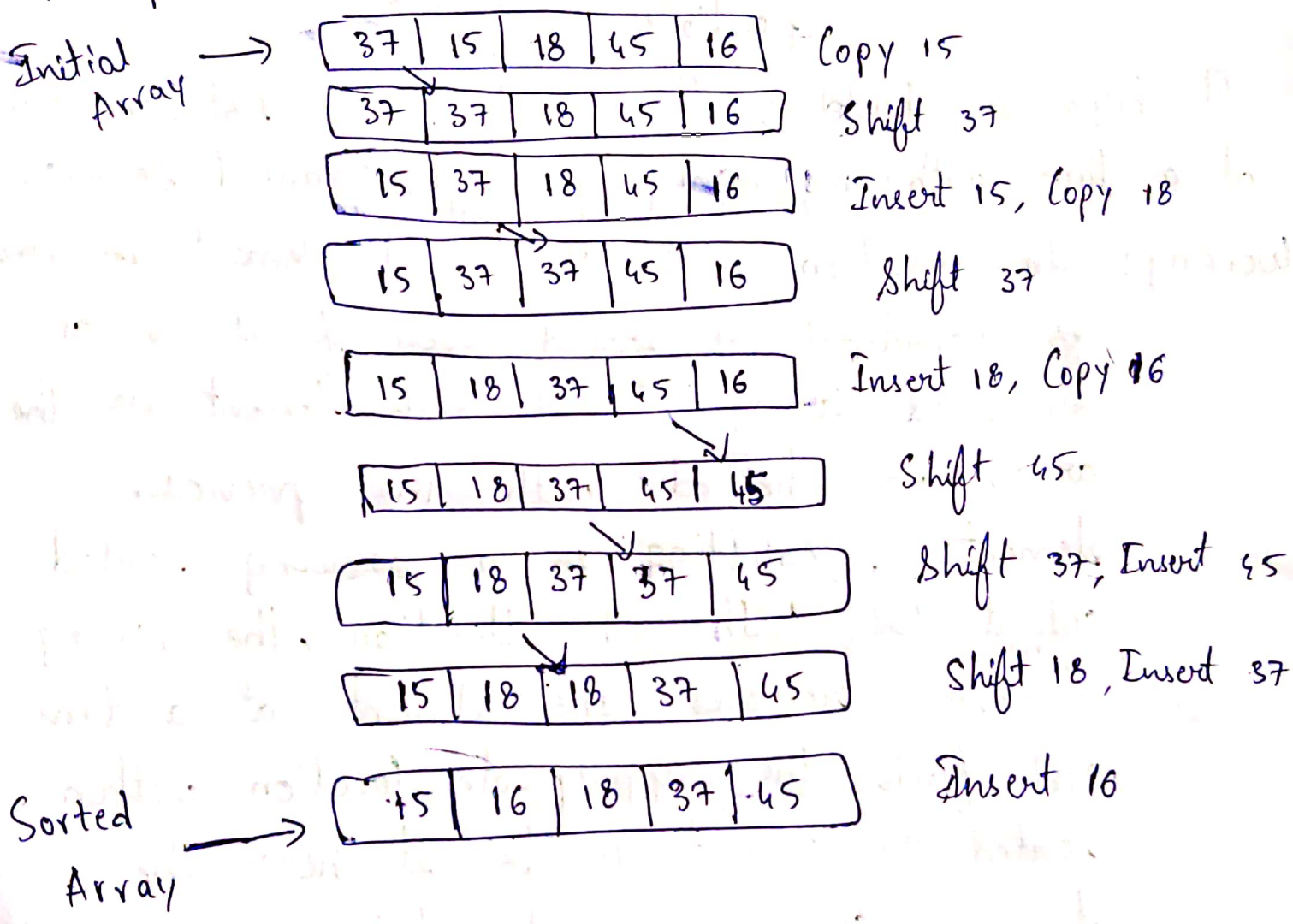
Ans: Insertion Sort - It is a sorting algorithm in which the elements are transfered one at a time to the right position.

It helps in building the final sorted list, one item at a time with movement of higher-ranked elements.

Working: In insertion sort, the first element in array is considered as sorted even if it is an unsorted array. In it, each element in the array is checked with the previous elements, resulting in a growing sorted output list. With each iteration, the sorting algorithm removes one element at a time and finds the appropriate location within sorted array and inserts it here. The iteration continues until whole array is sorted.

Time Complexity - Best Case - $O(n)$
Avg Case - $O(n^2)$
Worst Case - $O(n^2)$

Example: Consider array,



Selection Sort- It is a sorting algorithm which orders a array of values by repetitively putting a particular value into its final position.

Working:

(i) Find smallest value in array
(ii) Switch it with the value in first position
(iii) Find the next smallest value in array
(iv) Switch it with value in second position.
(v) This process is repeated till whole array is sorted.

Example: -Consider array, 15 2 35 3 47

Initial array → • 15 2 35 3 47 (2 is smallest)

• 2 15 35 3 47 (3 is next smallest to 2)

• 2 3 35 15 47 (15 is next smallest to 3)

Sorted array → • 2 3 15 35 47 (Sorted)

Complexity: $O(n^2)$

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

i. In alternate order

ii. Sum of elements in odd positions and product of elements in even positions.

iii. Elements which are divisible by m where m is taken from the user.

Ans. (I am writing 4 functions,
one for - Bubble sort
one for - i
one for - ii
one for - iii)

```
#include <stdio.h>
void bubblesort (int a[], int n)
{
    int i, j, temp;
    for(i=0; i<n; i++)
    {
      for(j=0; j<n-1; j++)
      {
          if(a[j] > a[j+1])
          {
              temp = a[j+1];
              a[j+1] = a[j];
              a[j] = temp;
          }
      }
    }
}
```

```
void printalternate(int a[], int n)
{
    int i;
    printf("Alternate elements in sorted array are:\n");
    for(i=0; i<n; i=i+2)
        printf("%d ", a[i]);
}

void sumoddproducteven(int a[], int n)
{
    int i, s=0, p=1;
    printf("Sum of odd position elements and
                product of even position elements are:");
    for(i=0; i<n; i++)
    {
        if(i%2==0)
            s=s+a[i];
        else
            p=p*a[i];
    }
    printf("%d ,%d", s, p);
}
void divisiblebym(int a[], int n)
{
    int i,m;
    printf("Enter m:");
    scanf("%d", &m);
    printf("Elements in sorted array divisible by m are:");
    for(i=0; i<n; i++)
    {
        if(a[i]%m==0)
            printf("%d ", a[i]);
    }
}
```

```
void main()
{
    int i,n;
    printf("Enter number of elements to sort:");
    scanf("%d",&n);
    int a[n];
    for(i=0; i<n; i++)
    {
        printf("Enter a[%d]:",i);
        scanf("%d",&a[i]);
    }
    bubblesort(a,n);
    printf("Sorted Data is:");
    for(i=0; i<n; i++)
        printf("%d",a[i]);
    printalternate(a,n);
    sumoddproducteven(a,n);
    divisiblebym(a,n);
}
```

Output:

```
Enter number of elements to sort:4
Enter a[0]:5
Enter a[1]:2
Enter a[2]:9
Enter a[3]:4
Sorted Data is: 2 4 5 9
Alternate elements in sorted array are: 2 5
Sum of odd position elements and product of
even position elements are: 7,36
Enter m: 2
Elements in array divisible by m are: 2 4
```

5.) Write a recursive program to implement in binary search.

Ans.

```
#include<stdio.h>
int bs_recursive(int a[], int start, int end, int element)
{
    int mid = (start + end)/2;
    if (start > end)
        return -1;
    if (a[mid] == element)
        return mid;
    else if (element < a[mid])
        bs_recursive(a, start, mid-1, element);
    else
        bs_recursive(a, mid+1, end, element);
}
void main()
{
    int n, i, k, start=0, end;
    printf("Enter number of elements to sort:");
    scanf("%d", &n);
    int a[n];
    end = n-1;
    for(i=0; i<n; i++)
    {
        printf("Enter a[%d]:", i);
        scanf("%d", &a[i]);
    }
    printf("Enter element to search:");
    scanf("%d", &k);
```

```
        printf ("%d found at position %d", k,
                bs_recursive(a,start,end,k)+1);
}
```

Output:

```
Enter number of elements to sort: 4
Enter a[0]: 12
Enter a[1]: 32
Enter a[2]: 11
Enter a[3]: 54
Enter element to search: 32
32 found at position 3
```