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 the array where the element is asked from user
b. Ask the user to enter any two locations print the sum and product of values at those locations in the sorted array

```
#include <stdio.h>
int binarySearch(int arr[], int a, int b, int x)
{
   if (b >= a) {
      int mid = a + (b - a) / 2;
      if (arr[mid] == x)
         return mid;
      if (arr[mid] > x)
         return binarySearch(arr, a, mid - 1, x);
      return binarySearch(arr, mid + 1, b, x);
   }
   return -1;
}
int main()
{
   int num;
   printf("Enter the size of array : ");
   scanf("%d",&num);
   int i,j,a,val[num],op,var,p1,p2,sum,pro;
   for(a=0;a<num;a++)
   {
      printf("Enter Value : ");
      scanf("%d",&val[a]);
   }
   for (i = 0; i < num; ++i)
   {
      for (j = i + 1; j < num; ++j)
      {
         if (val[i] < val[j])
         {
            a = val[i];
            val[i] = val[j];
            val[j] = a;

         }
      }
```

```
    }
    printf("Array in descending order : ");
    for(i=0;i<num;i++)
    {
        printf("%d",val[i]);
    }
    printf("\n**OPERATION_LIST**\n");
    printf("1.Find value at entered position\n2.Find the position of element\n3.Printing
sum&multiplication of values at entered positions");
    printf("\nEnter Choice : \n");
    scanf("%d",&op);
    switch(op)
    {
        case 1:
        printf("Enter the position to obtain value :");
        scanf("%d",&var);
        printf("The value at %d position is %d",var,val[var]);
        break;
        case 2:
        printf("Enter element to find position : ");
        scanf("%d",&var);
        int result = binarySearch(val, 0, num - 1, var);
        (result == -1) ? printf("Element is not present in array")
        :printf("Element is present at index %d",result);
        return 0;
        case 3:
        printf("\nEnter two positions to find sum and product of values\n");
        scanf("%d %d",&p1,&p2);
        sum=val[p1]+val[p2];
        pro=val[p1]*val[p2];
        printf("SUM=%d\n",sum);
        printf("MULTIPLICATION=%d",pro);
        break;
    }

}
```

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

```
#include<stdlib.h>
#include<stdio.h>
void merge(int arr[], int l, int m, int r)
```

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

    /* create temp arrays */
    int L[n1], R[n2];

    /* Copy data to temp arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1+ j];

    /* Merge the temp arrays back into array*/
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = l; // Initial index of merged subarray
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    /* Copy the remaining elements of L[] if any*/
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }

    /* Copy the remaining elements of R[], if any */
    while (j < n2)
```

```
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}

/* Function to print an array */
void printArray(int A[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

int main()
{
    int siz,v;
    printf("Enter array size : ");
    scanf("%d",&siz);
    int val[siz];
    for(v=0;v<siz;v++)
    {
        printf("Enter Value :");
        scanf("%d",&val[v]);
    }
    printf("Given array is \n");
    printArray(val,siz);
    mergeSort(val, 0, siz-1);
```

```
    printf("\nSorted array is \n");
    printArray(val,siz);
    int k,f,l,p1,p2,temp;
    printf("Enter the value of k to find the product of elements from first and last : ");
    scanf("%d",&k);
    p1=p2=1;
    for(f=0;f<=k;f++)
    {
        temp=val[f];
        p1*=temp;
    }
    for(l=siz-1;l>=k;l--)
    {
        temp=val[l];
        p2*=temp;
    }
    printf("Product of kth elements from first and last are : %d %d",p1,p2);
}
```

3. Discuss Insertion sort and Selection sort with examples

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

```
#include <stdio.h>
/*Bubblesort function*/
void bubbleSort(int ar[],int n)
{
   int i,j,temp;
   for (i = 0; i < n-1; i++)
   for (j = 0; j < n-i-1; j++)
   if (ar[j] > ar[j+1])/*Exchanging values using condition and temp variable*/
   {
     temp=ar[j];
     ar[j]=ar[j+1];
     ar[j+1]=temp;
   }
}
```

```
int main()
{
    int siz,i;
    printf("Enter size of required array : ");
    scanf("%d",&siz);
    int arr[siz];
    for(i=0;i<siz;i++)
    {
        printf("Enter element : ");
        scanf("%d",&arr[i]);
    }
    bubbleSort(arr,siz);
    printf("Sorted array: \n");
    for(i=0;i<siz;i++)
    {
        printf("%d",arr[i]);
        printf("\t");
    }
    printf("\n/**MENU**/\n");
    printf("1.Display elements in alternate order\n");
    printf("2. Sum of elements in odd positions and Product of elements in even positions\n");
    printf("3. Divisible by m\n");
    int op,sum=0,product=1,m;
    printf("Enter Choice : ");
    scanf("%d",&op);
    switch(op)
    {
        case 1:
        for(i=0;i<siz;i+=2)
        {
            printf("%d\t",arr[i]);
        }
        case 2:
        for(i=0;i<siz;i+=2)
        {
            sum=sum+arr[i];
        }
        for(i=1;i<siz;i+=2)
        {
            product=product*arr[i];
        }
        printf("Sum : %d\n",sum);
        printf("Product : %d\n",product);
```

```
        case 3:
        printf("Enter value m :");
        scanf("%d",&m);
        printf("Numbers divisble by %d are :\n",m);
        for(i=0;i<siz;i++)
        {
            if(arr[i]%m==0)
            {
                printf("%d\t",arr[i]);
            }
        }
    }
}
```

5.Write a recursive program to implement binary search?

```
#include <stdio.h>
int binarysearch(int a[],int l,int h,int x)
{
    int mid = (l + h) / 2;
    if (l>h) return -1;
    if (a[mid] == x)
    return mid;
    if (a[mid] < x)
    return binarysearch(a, mid+1,h,x);
    else
    return binarysearch(a,l,mid-1,x);
}

int main(void)
{
  int a[100];
  int siz,pos,val;

  printf("Enter length of the array");
  scanf("%d", &siz);

  printf("\nEnter array elements\n");
  for(int i=0; i<siz; i++)
    scanf("%d", &a[i]);

  printf("Enter element to search\n");
  scanf("%d", &val);

  pos = binarysearch(a,0,siz-1,val);
```

```
  if (pos < 0 )
    printf("Cannot find the element %d in the array.\n",val);
    else
    printf("The position of %d in the array is %d.\n",val,pos+1);
return 0;
}
```