N. Lakshmi Swarna
Durga (CSE-G)
AP19110010334.

1) code:

```
#include<stdio.h>
void sort(int arr[], int n)
{ int i,j,swap;
  for(i=0;i<n-1;i++)
  {
    for(j=0;j<n-i-1;j++)
    {
      if(arr[j]<arr[j+1])
      { swap=arr[j]
        arr[j]=arr[j+1]
        arr[j+1]=swap;
      }
    }
  }
}
int main
{ int i, j, low, high, middle, n, search, swap;
  printf("Enter number of elements:");
  scanf("%d", &n);
  int array[n];
  priutf("Enter elements: \n");
  for(i=0; i<n; i++)
      scanf("%d", &array[i]);
  priutf("array is in descending orde
  sort(array, n);
  for(i=0; i<n; i++)
  { priutf("%d", array[i]);
  }
```

```
printf("Enter value to be searched");
scanf("%d", &search);
low=0
high=n-1
while(low<=h)
{   middle=(low+high)/2
    if(array[middle]<search)
        low=middle+1
    else if(array[middle]==search)
    {  printf("%d found at location %d",
                search, middle+1);
        break;
    }
    else
        high=middle-1;
}
if(low>high)
    printf("Not found");
return 0;
}
```

Output:

Enter number of elements: 3

2

1

3

The descending order is 3 2 1.

Enter element to be searched: 3

3 found at 1

```
#include<stdio.h>
#define ms 100
int a[ms];
void merge(int l1,int u1,int l2,int u2)
{
  int i,j,k,temp[ms];
  k=0
  i=l1
  j=l2;
while((i<=u1)&&(j<=u2){
   if(a[i]<a[j]{
     temp[k]=a[i];i++;k++;
}
else{
    temp[k]=a[j];j++;k++;
  }
}
while(j<=u2){
    temp[k]=a[j];j++;k++
}
for(i=l1;k=0;i<=u2;i++,k++){
    a[i]=temp[k];
  }
}
void mergesort(int lb,int ub){
  if(lb<ub)
    { int mid=(ub+lb)/2;
      mergesort(lb,mid);
      mergesort(mid+1,ub)
      merge(lb,mid,mid+1,ub);
}
```

```
int main()
    int i, n, product = 1, k;
    printf("array size:");
    scanf("%d", &n);
    for(i=0; i<n; i++)
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    mergesort(0, n-1)
    printf("Enter k");
    scanf("%d", &k);
    for(i=0; i<k; i++) {
        product *= a[i];
    }
    printf("\n product %d\n", product)
    return 0;
}
```

```
Enter elements: 4
Enter elements:
2
3
1
4
```

Array in descending order: 4 3 2 1

Sorted array: 1 2 3 4

Enter k: 4

product: 20

4th is lowest
1 2 4 3
After this process continues and eventually swaps with least number with respect to its position
1 2 3 4

Code:

```
#include<stdio.h>
void sort (int arr[ ],int n)
{  int i,j,swap;
    for(i=0; i<n-1;i++)
    {  for(j=0; j<n-i-1;j++)
       {  if(arr[j]<arr[j+1])
          {  swap = arr[j]
             arr[j] = arr[j+1]
             arr[i+1]=swap
          }
       }
    }
}
void print (int arr[],int n)
{  int i;
    printf("\order:\n");
    for(i=0;i<=n;i+=2)
       printf("%d\t",arr[i]);
}
void calculation(int arr[],int k)
{  int i,sum=0,product=1;
    for(i=0;i<k;i++)
    {  if((i+1)%2==0)
       product*=arr[i];
    else
       sum+=arr[i]
```

Insertion sort:
One element from the array is selected and is composed to one side of a different array and inserted to the proper position while shifting the rest of elements accordingly

let the unsorted array

3	1	4	2

Insertion sort first compress two elements

1	3	4	2

→

1	3	4	2

→ The first element is already small number compared to the second number so the order doesnot changes
→ In the second case 3,4 are in ascending order so order doesnot changes.
This order follows till the end
1 3 2 4
And the process continues till end
1 2 3 4

Selection sort
It is basically selection of elements position from the start with the other rest of the lements.
Example
3 1 4 2
For the first position in the sorted list, the whole list is scanned sequentially. The first position where a^2 is stored presently, we search the whole list and find that

```
printf("\n sum of elements : %d ", sum, product);
void divisiable(int arr[], int x, int k)
{ int i;
   printf("Elements divided by %d ", k);
   for(i=0; i<x; i++)
   { if(arr[i]%k==0)
        printf("%d\t", arr[i]);
   }
}
int main()
{ int i,n,m;
   printf("Enter no of elements");
   scanf("%d", &n);
   int array[n];
   printf("Enter elements: \n");
   for(i=0; i<n; i++)
      scanf("%d", &array[i]);
   printf("Array is in descending order");
   sort(array,n);
   for(i=0; i<n; i++)
      printf("%d\t", array[i]);
   calculation(array,n)
   printf("Enter m:");
   scanf("%d", &m);
   divisible(array,n,m);
   return 0
```

output:

Enter no of elements: 5

Enter elements

1

2

4

3

5

descending order : 5 4 3 2 1

sum of elements odd positions: 9
sum of even positions: 8
Value of a: 2
Divisible elements: 4 2

code:

```
#include<stdio.h>
void binary_search(int[],int,int,int)
void bubblesort(int[],int);
int main()
{ int key,size,i;
  int list[25];
  printf("size of list:");
  scanf("%d",&size);
  printf("enter elements:");
  for(i=0;i<size;i++)
      scanf("%d",&list[i]);
  bubblesort(list,size);
  printf("\n");
  printf("Enter search key:");
  binarysearch(list,0,size,key);
}
void bubblesort(int list[],int size)
{ int temp,i,j;
  for(i=0,i<size;i++).
  { for(j=i;j<size;i++)
    { if(list[i]>list[j])
      { temp=list[i];
        list[i]=list[j]
        list[j]=temp
      }
    }
  }
```

```
void binarysearch (int list[], int first, int last,
                   int key)
{ int mid;
   if (first > last)
   {   printf ("Number not found!\n");
       return;
   }
   mid = (first+last)/2
   if (list [mid] == key)
   {  printf ("Number found!\n");
   }
   else if (list [mid] > key)
       binarysearch (list, last, mid - 1, key);
    else if (list [mid] < key)
       binary search (list, mid + 1, last, key)
}
```

Output

```
Enter size: 4
1
2
3
4
Enter key search: 4
Number found.
```