1)

```
#include<stdio.h>
 void main()
 {
 int  arra[100],i,n,x,f,l,m,flag=0;
 printf("Input no. of elements in  an array\n");
 scanf("%d",&n);
 printf("Input  %d value in ascending order\n",n);
 for(i=0;i<n;i++)
 scanf("%d",&arra[i]);
 printf("Input  the value to be search : ");
 scanf("%d",&x);
 /* Binary Search  logic */
 f=0;l=n-1;
 while(f<=l)
 {
 m=(f+l)/2;
 if(x==arra[m])
 {
 flag=1;
 break;
 }
 else  if(x<arra[m])
 l=m-1;
 else
 f=m+1;
 }
 if(flag==0)
 printf("%d  value not found\n",x);
 else
```

```
 printf("%d value  found at %d position\n",x,m);
 }
```

2)

```
#include <stdio.h>
 void main()
 {
 int arr[10];
 int i, j, N, temp;
/* function  declaration */
 int find_max(int  b[10], int k);
 void exchang(int  b[10], int k);
 printf("\nInput no. of values in the array : ");
 scanf("%d",&N);
  printf("\nInput  the elements one by one: ");
 for(i=0; i<N ;  i++)
 {
 scanf("%d",&arr[i]);
 }
 /* Selection sorting  begins */
 exchang(arr,N);
 printf("Sorted  array :\n");
 for(i=0; i< N ;  i++)
 {
 printf("%d\n",arr[i]);
 }
}
/* function to find the maximum value */
 int find_max(int b[10], int k)
 {
```

```
int max=0,j;
for(j = 1; j <=  k; j++)
{
if ( b[j] >  b[max])
{
max =  j;
}
}
return(max);
}
void exchang(int  b[10],int k)
{
int  temp, big, j;
for ( j=k-1;  j>=1; j--)
{
big =  find_max(b,j);
temp = b[big];
b[big] = b[j];
b[j] = temp;
}
return;
}
```

3)

```
#include <stdio.h>
void main()
{
int arr[10];
int i, j, N, temp;
/* function  declaration */
```

```
 int find_max(int  b[10], int k);
 void exchang(int  b[10], int k);
 printf("\nInput no. of values in the array: ");
 scanf("%d",&N);
printf("\nInput the elements: ");
 for(i=0; i<N ;  i++)
 {
 scanf("%d",&arr[i]);
 }
 /* Selection sorting  begins */
 exchang(arr,N);
 printf("Sorted  array :\n");
 for(i=0; i< N ;  i++)
 {
 printf("%d\n",arr[i]);
 }
}
/* function to find the maximum value */
 int find_max(int b[10], int k)
 {
 int max=0,j;
 for(j = 1; j <=  k; j++)
 {
 if ( b[j] >  b[max])
 {
 max =  j;
 }
 }
 return(max);
 }
```

```
void exchang(int  b[10],int k)
{
int  temp, big, j;
for ( j=k-1;  j>=1; j--)
{
big =  find_max(b,j);
temp = b[big];
b[big] = b[j];
b[j] = temp;
}
return;
}
```

4)

```
#include<stdio.h>
void main()
{
int  arra[10],i,j,n,array_key;
printf("Input  no. of values in the array: \n");
scanf("%d",&n);
printf("Input  %d array value(s): \n",n);
for(i=0;i<n;i++)
scanf("%d",&arra[i]);

/* Insertion Sort  */
for (i = 1; i <  n; i++)
{
array_key =  arra[i];
j = i-1;
```

```
 while (j >=  0 && arra[j] > array_key)
 {
 arra[j+1] = arra[j];
 j = j-1;
 }
 arra[j+1] =  array_key;
 }
 printf("Sorted  Array: \n");
 for (i=0; i <  n; i++)
 printf("%d  \n", arra[i]);
 }
```

5)

```
#include<stdio.h>
/* Function to merge the two haves arra[l..m] and arra[m+1..r] of array
arra[] */
 void merge(int arra[], 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] = arra[l + i];
 for(j = 0; j < n2; j++)
 R[j] = arra[m + 1+ j];
 i = 0;
 j = 0;
```

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

```
 if (l < r)
 {
 int m = l+(r-l)/2; //Same as (l+r)/2, but avoids overflow for large l and
h
 mergeSort(arra, l, m);
 mergeSort(arra, m+1, r);
 merge(arra, l, m, r);
 }
 }
/* Function to print an array */
 void print_array(int A[], int size)
 {
 int i;
 for (i=0; i < size; i++)
 printf("%d ", A[i]);
 printf("\n");
 }
/* Test above functions */
 int main()
 {
 int arra[] = {125, 181, 130, 25, 61, 887};
 int arr_size = sizeof(arra)/sizeof(arra[0]);
 printf("Given array is \n");
 print_array(arra, arr_size);
 mergeSort(arra, 0, arr_size - 1);
 printf("\nSorted array is \n");
 print_array(arra, arr_size);
 return 0;
 }
```

6)

```
#include <stdio.h>
 void main()
 {
 int arr[10], no,  i, j, c, heap_root, temp;
 printf("Input number of elements: ");
 scanf("%d", &no);
 printf("\nInput array values one by one : ");
 for (i = 0; i <  no; i++)
 scanf("%d", &arr[i]);
 for (i = 1; i <  no; i++)
 {
 c = i;
 do
 {
 heap_root  = (c - 1) / 2;
 /* to create MAX arr  array */
 if  (arr[heap_root] < arr[c])
 {
 temp =  arr[heap_root];
 arr[heap_root] = arr[c];
 arr[c]  = temp;
 }
 c =  heap_root;
 } while (c !=  0);
 }

 printf("Heap  array : ");
 for (i = 0; i <  no; i++)
 printf("%d\t ", arr[i]);
```

```
for (j = no - 1; j  >= 0; j--)
{
temp = arr[0];
arr[0] = arr[j];
arr[j] = temp;
heap_root = 0;
do
{
c = 2 *  heap_root + 1;
if  ((arr[c] < arr[c + 1]) && c < j-1)
c++;
if  (arr[heap_root]<arr[c] && c<j)
{
temp =  arr[heap_root];
arr[heap_root] = arr[c];
arr[c]  = temp;
}
heap_root  = c;
} while (c  < j);
}
printf("\nSorted  array : ");
for (i = 0; i <  no; i++)
printf("\t%d", arr[i]);
printf("\n");
}
```

7)

```
#include<stdio.h>
 int  n;
 void  main()
```

```
{
int arr[30],l,r,i;
void quick_sort(int arr[],int,int);
printf("\nInput number of elements:  ");
scanf(" %d",&n);
printf("\nInput  array values one by one: ");
for(i=0;i<n;i++)
scanf(" %d",&arr[i]);
l=0; r=n-1;
quick_sort(arr,l,r);
printf("\nThe quick sorted array is:  ");
for(i=0;i<n;i++)
printf(" %d",arr[i]);
printf("\n");
}
void quick_sort(int arr[],int low,int high)
{
int temp,left,right,x,k;
if(low>=high)
return;
else
{
x=arr[low];
right=low+1;
left = high;
while(right<=left)
{
while(arr[right]<x && right  <= high)
{
right ++;
```

```
 }
 while(arr[left]>x && left >  low)
 {
 left--;
 }
 if(right<left)
 {
 temp=arr[right];
 arr[right]=arr[left];
 arr[left]=temp;
 right++;
 left--;
 }
 }
 arr[low]=arr[left];
 arr[left]=x;
 quick_sort(arr,low,left-1);
 quick_sort(arr,left+1,high);
 }
 }
```

8)

```
#include  <stdio.h>

 void print(int *a, int n) {
 int i;
 for (i = 0; i < n;  i++)
 printf("%d\t", a[i]);
 }
```

```
void radix_sort(int *a, int n) {
int i, b[10], m = 0,  exp = 1;
for (i = 0; i < n;  i++) {
if (a[i] > m)
m = a[i];
}

while (m / exp >  0) {
int box[10] = { 0 };
for (i = 0; i <  n; i++)
box[a[i] / exp %  10]++;
for (i = 1; i <  10; i++)
box[i] += box[i -  1];
for (i = n - 1; i  >= 0; i--)
b[--box[a[i] / exp  % 10]] = a[i];
for (i = 0; i <  n; i++)
a[i] = b[i];
exp *= 10;
}
}

int main() {
int arr[10];
int i, num;

printf("Input  number of elements: ");
scanf("%d",  &num);

printf("\nInput  array elements one by one : ");
for (i = 0; i <  num; i++)
```

```
 scanf("%d", &arr[i]);

 printf("\nArray  elements : ");
 print(&arr[0],  num);

 radix_sort(&arr[0], num);

 printf("\nSorted  elements : ");
 print(&arr[0],  num);

 return 0;
 }
```

9)

```
#include  <stdio.h>

 /*  Counting sort function  */
 void  counting_sort(int a[], int k, int n)
 {
 int i, j;
 int b[15], c[100];
 for (i = 0; i <= k; i++)
 c[i] = 0;
 for (j = 1; j <= n; j++)
 c[a[j]] = c[a[j]] + 1;
 for (i = 1; i <= k; i++)
 c[i] = c[i] + c[i-1];
 for (j = n; j >= 1; j--)
 {
 b[c[a[j]]] = a[j];
```

```
c[a[j]] = c[a[j]] - 1;
}
printf("The Sorted array is : ");
for (i = 1; i <= n; i++)
printf("%d,", b[i]);
}
int  main()
{
int n, k = 0, a[15], i;
printf("Input number of elements:  ");
scanf("%d", &n);
printf("Input the array elements one  by one: \n");
for (i = 1; i <= n; i++)
{
scanf("%d", &a[i]);
if (a[i] > k) {
k = a[i];
}
}
counting_sort(a, k, n);
printf("\n");
return 0;
}
```