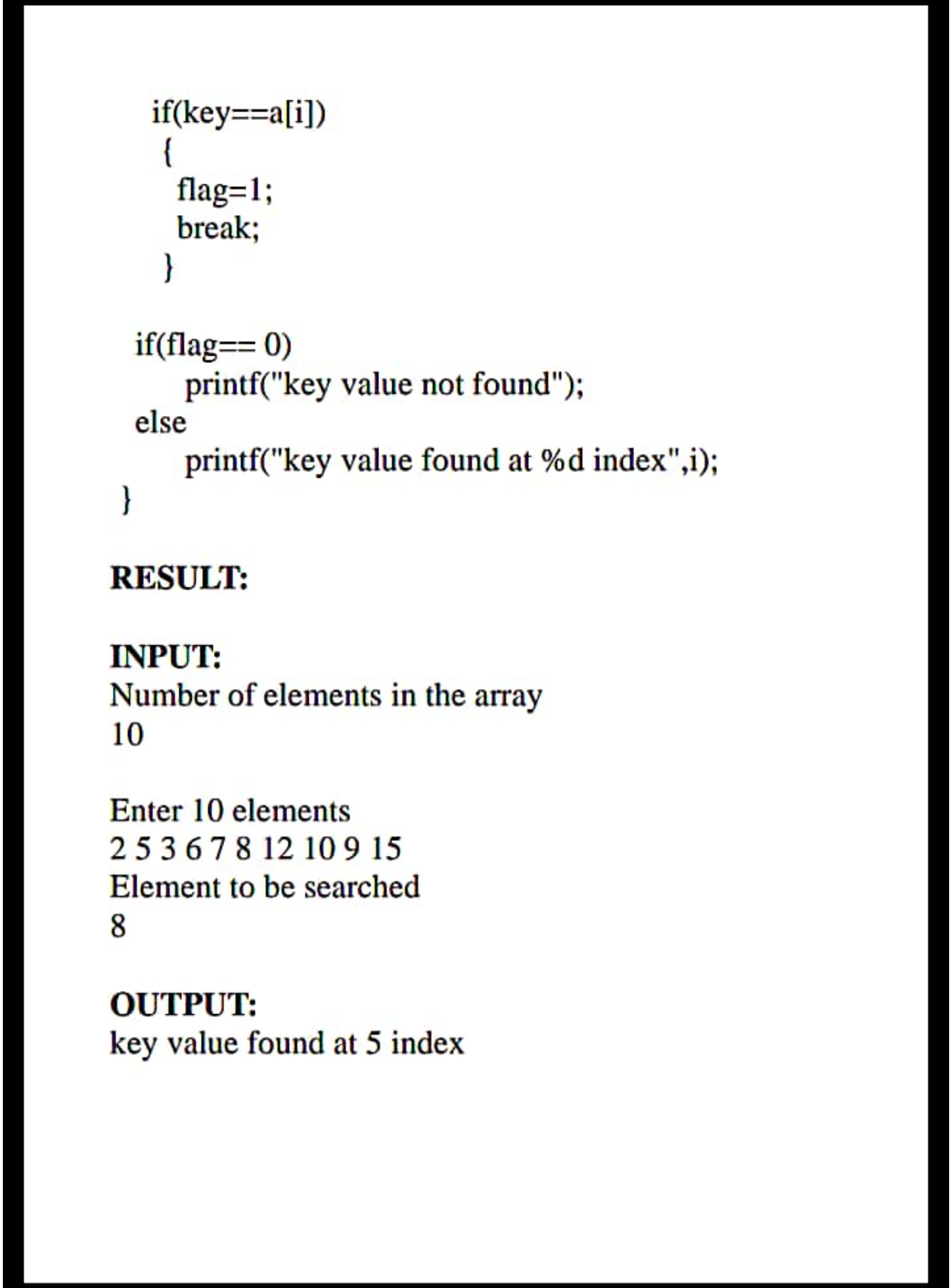
Sorting and Searching:

a. Write a C program that uses non recursive function to search for a Key value in a given list of integers using linear search method.

```
#include<stdio.h>

int lsearch(int a[],int,int);

int main()
 {
  int i,j,a[100],key,n;

  printf("Number of elements in the array\n");
  scanf("%d",&n);

  printf("\nEnter %d elements\n",n);

  for(i=0;i<n;i++)
     scanf("%d",&a[i]);

  printf("Element to be searched\n");
  scanf("%d",&key);

  lsearch(a,n,key);
 }

int lsearch(int a[],int n,int key)
 {
  int flag=0,i;
  for(i=0;i<n;i++)
```

```
     if(key==a[i])
      {
       flag=1;
       break;
      }
```



```
    if(key==a[i])
     {
       flag=1;
       break;
     }

  if(flag== 0)
       printf("key value not found");
  else
       printf("key value found at %d index",i);
}
```

RESULT:

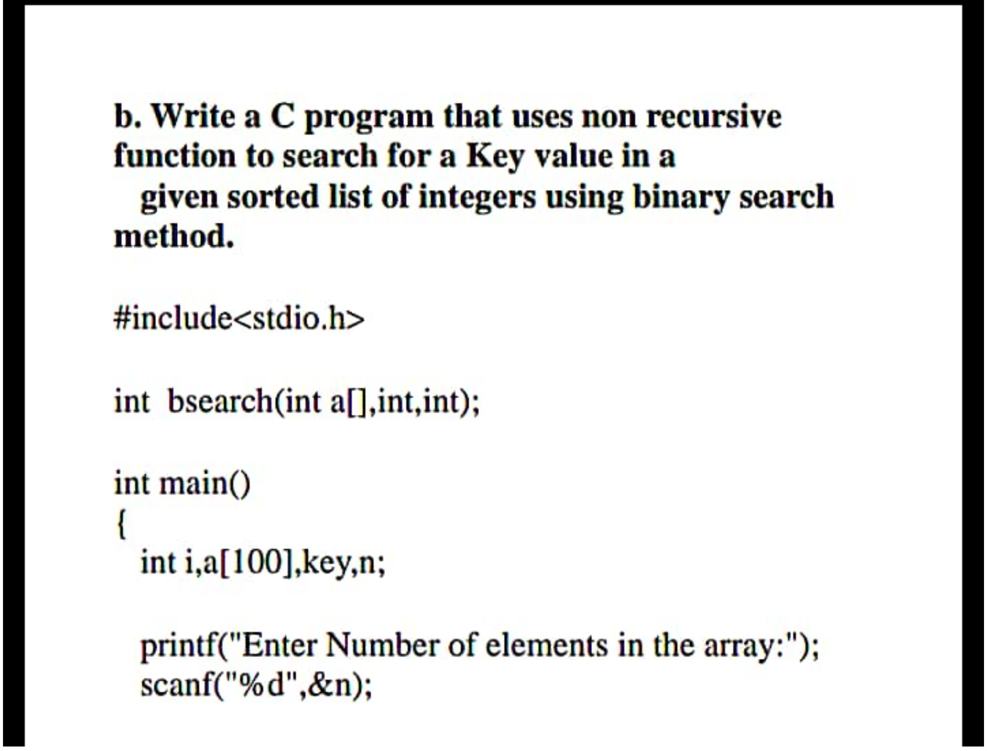
INPUT:

Number of elements in the array
10

Enter 10 elements
2 5 3 6 7 8 12 10 9 15
Element to be searched
8

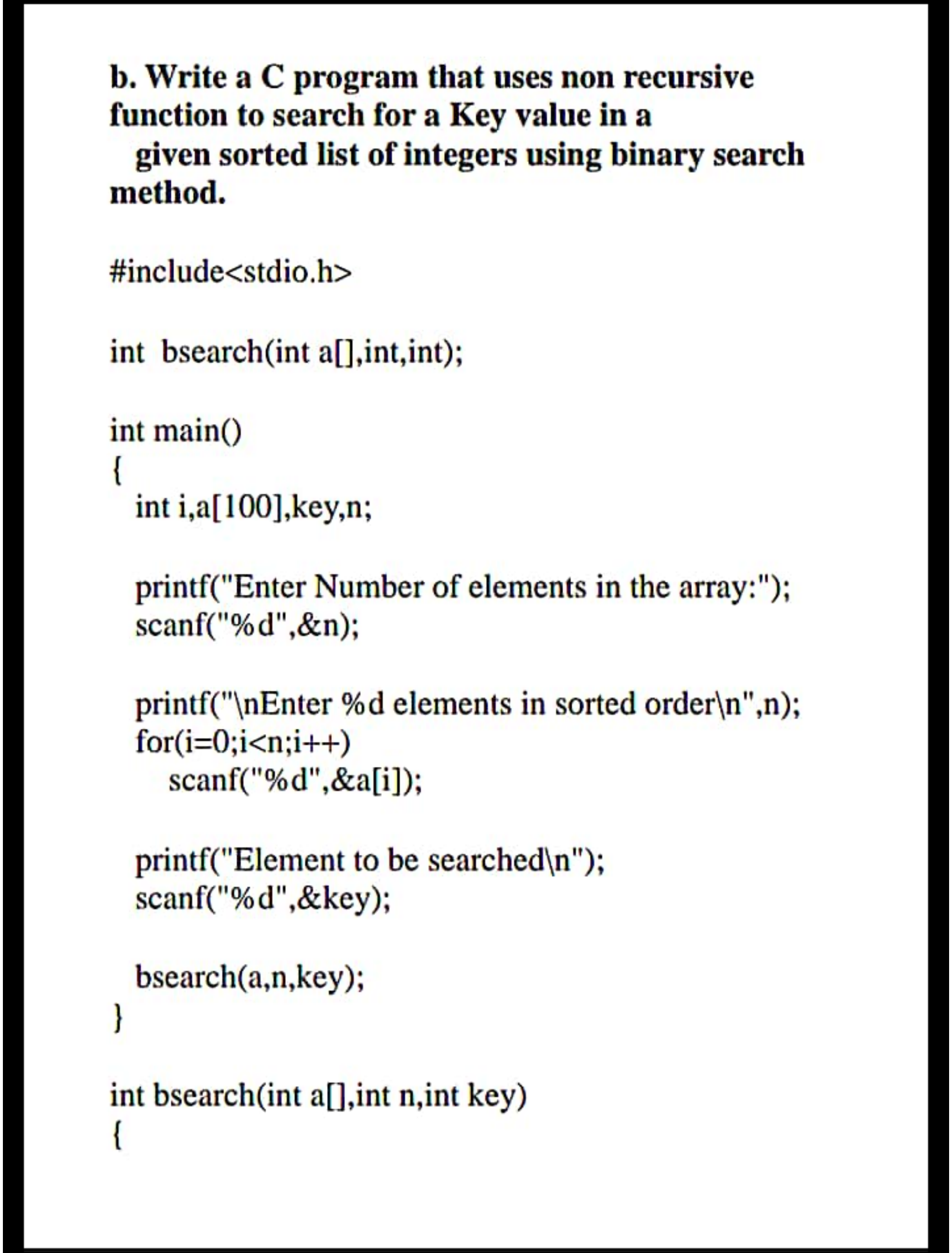
OUTPUT:

key value found at 5 index



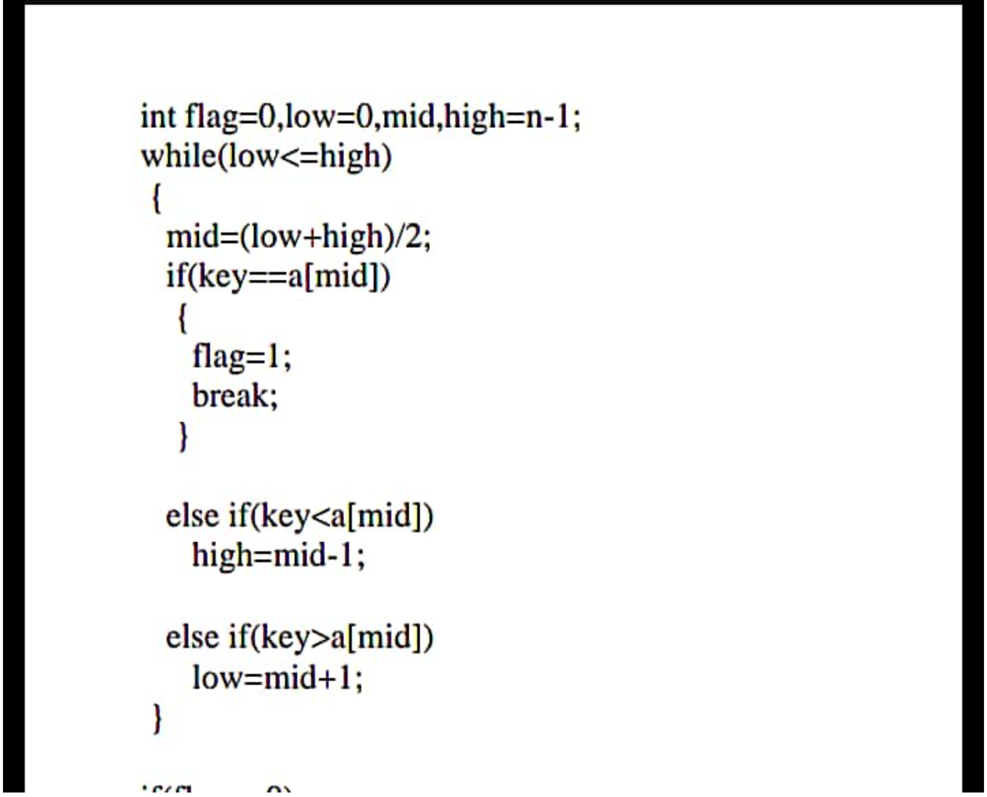
b. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.

```
#include<stdio.h>

int  bsearch(int a[],int,int);

int main()
{
  int i,a[100],key,n;

  printf("Enter Number of elements in the array:");
  scanf("%d",&n);
```

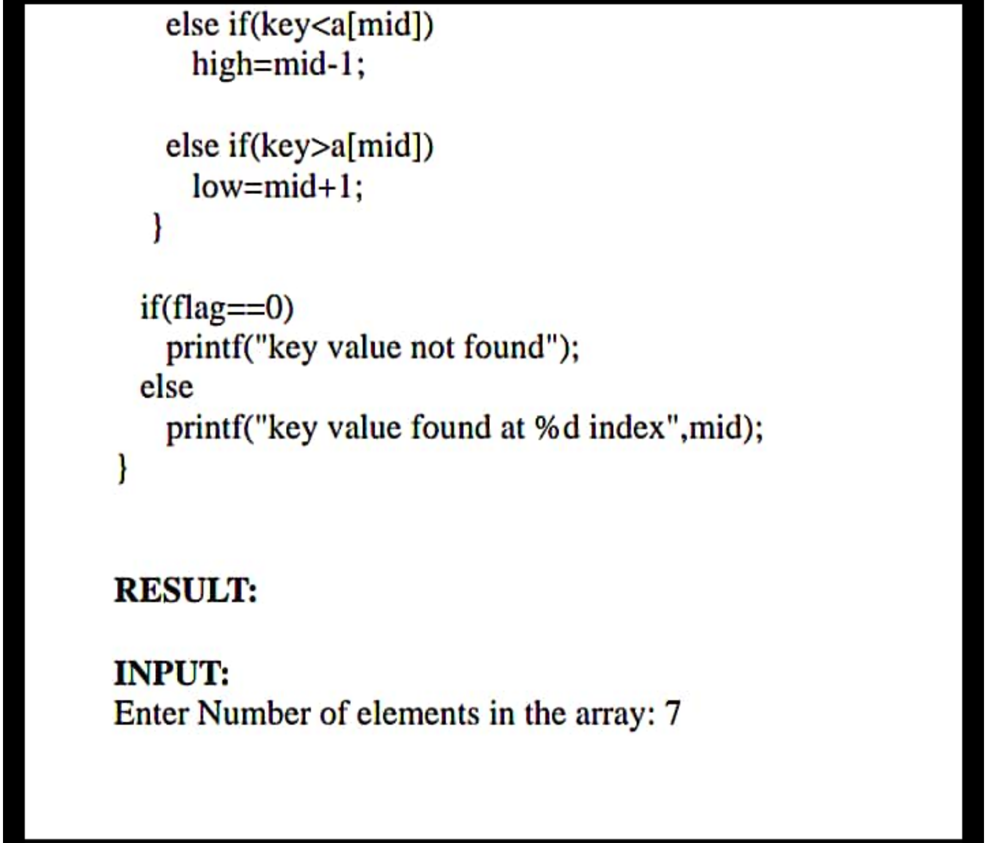


b. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using binary search method.

```
#include<stdio.h>

int  bsearch(int a[],int,int);

int main()
{
  int i,a[100],key,n;

  printf("Enter Number of elements in the array:");
  scanf("%d",&n);

  printf("\nEnter %d elements in sorted order\n",n);
  for(i=0;i<n;i++)
     scanf("%d",&a[i]);

  printf("Element to be searched\n");
  scanf("%d",&key);

  bsearch(a,n,key);
}

int bsearch(int a[],int n,int key)
{
```



```
  int flag=0,low=0,mid,high=n-1;
  while(low<=high)
   {
    mid=(low+high)/2;
    if(key==a[mid])
     {
      flag=1;
      break;
     }

    else if(key<a[mid])
      high=mid-1;

    else if(key>a[mid])
      low=mid+1;
   }

  if(flag==0)
```

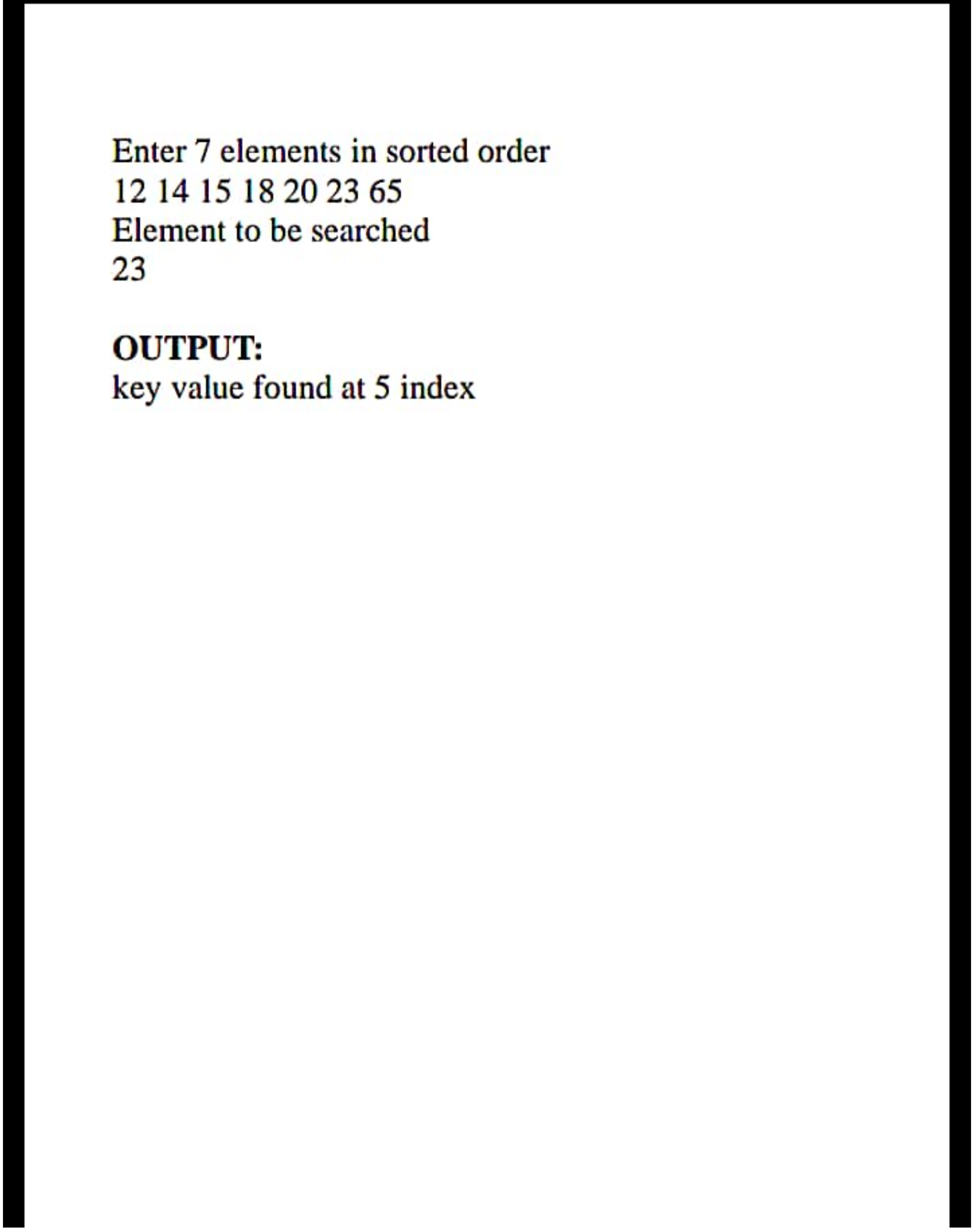


```
      else if(key<a[mid])
        high=mid-1;

      else if(key>a[mid])
        low=mid+1;
    }

  if(flag==0)
      printf("key value not found");
  else
      printf("key value found at %d index",mid);
}
```

RESULT:

INPUT:

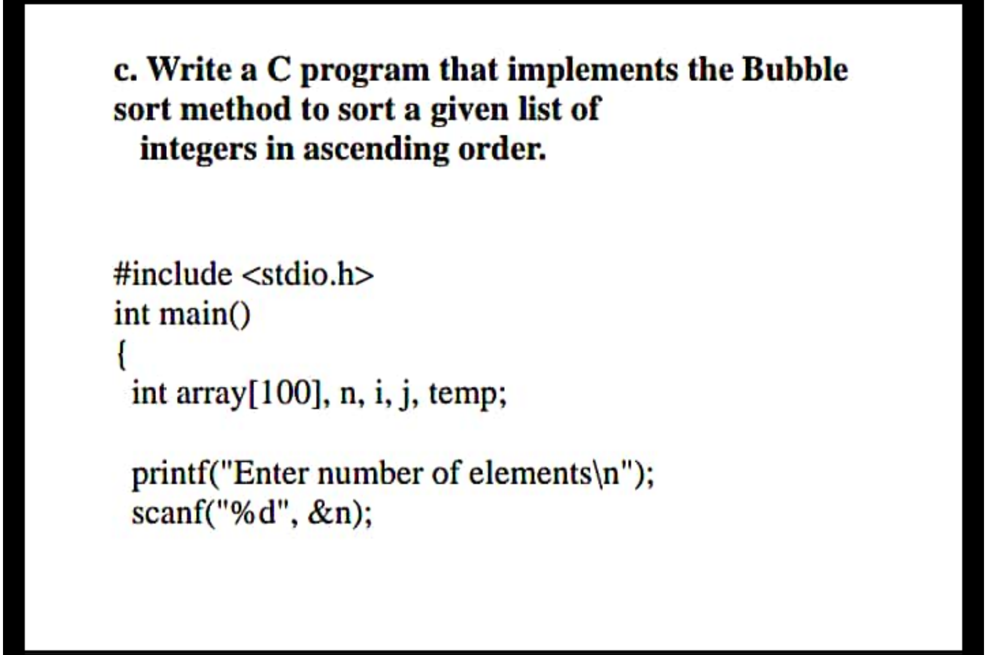
Enter Number of elements in the array: 7



```
Enter 7 elements in sorted order
12 14 15 18 20 23 65
Element to be searched
23
```

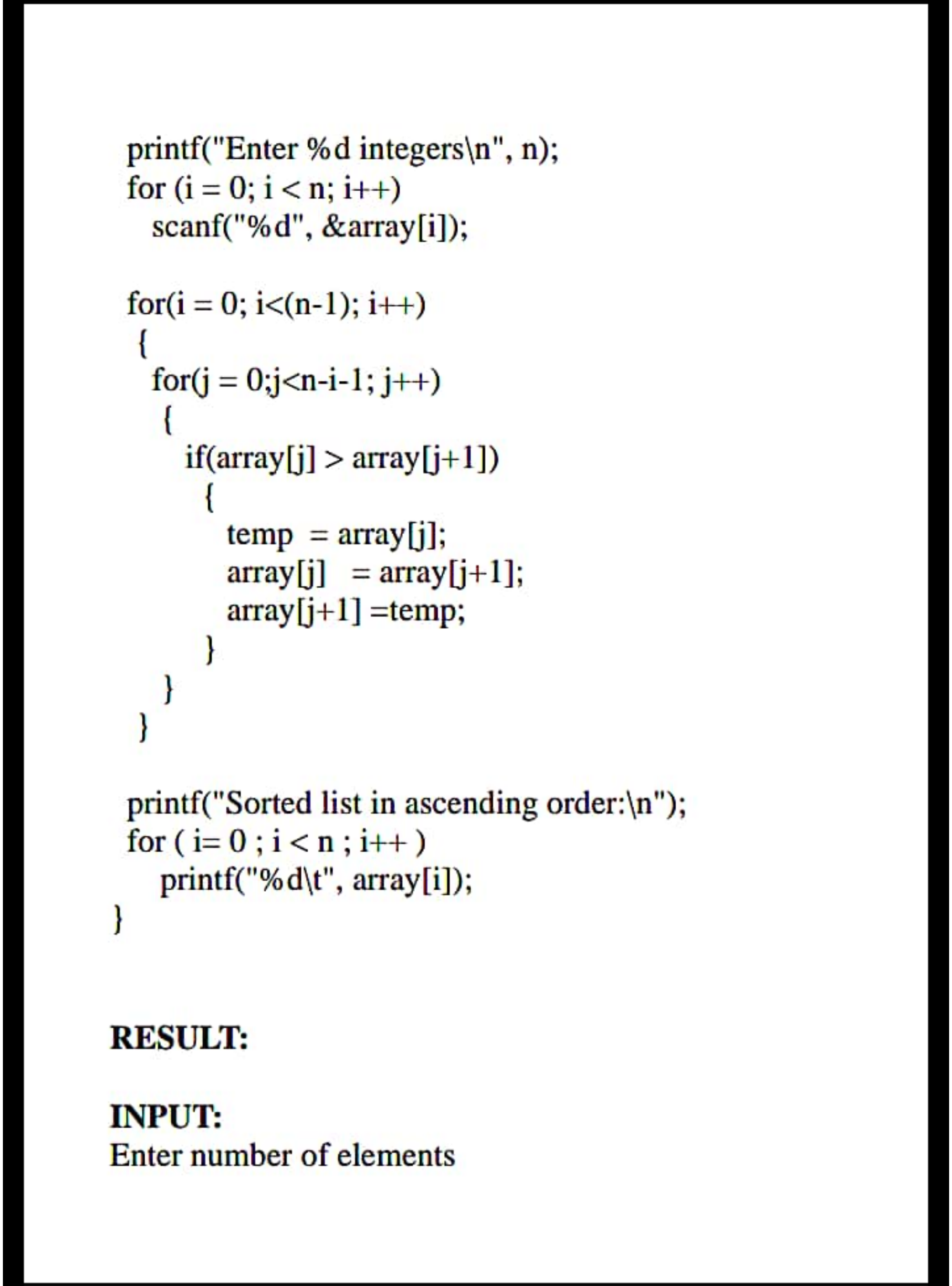
OUTPUT:

```
key value found at 5 index
```



c. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

```
#include <stdio.h>
int main()
{
  int array[100], n, i, j, temp;

  printf("Enter number of elements\n");
  scanf("%d", &n);
```



```
  printf("Enter %d integers\n", n);
  for (i = 0; i < n; i++)
    scanf("%d", &array[i]);

  for(i = 0; i<(n-1); i++)
   {
    for(j = 0;j<n-i-1; j++)
     {
       if(array[j] > array[j+1])
        {
          temp  = array[j];
          array[j]   = array[j+1];
          array[j+1] =temp;
        }
     }
   }

  printf("Sorted list in ascending order:\n");
  for ( i= 0 ; i < n ; i++ )
     printf("%d\t", array[i]);
}
```

RESULT:

INPUT:
Enter number of elements

RESULT:

INPUT:

```
Enter number of elements
10
Enter 10 integers
12 1 4 7 23 76 45 21 54 47
```

OUTPUT:

```
Sorted list in ascending order:
1      4      7      12      21      23      45      47      54
76
```

d. Write a C program that sorts the given array of integers using selection sort in descending order.

```
#include <stdio.h>

int main()
{
      int a[10] = { 3,4,7,6,5,1,2,8,10,9 };
      int n = 10;
      printf("\nArray Data : ");

      for (int i = 0; i < n; i++)   //Loop for displaying the
data of array
          {
            printf(" %d ", a[i]);
          }

        for (int i = 0; i < n; i++)   //Loop for descending
ordering
          {
            for (int j = 0; j < n; j++)
              {
```

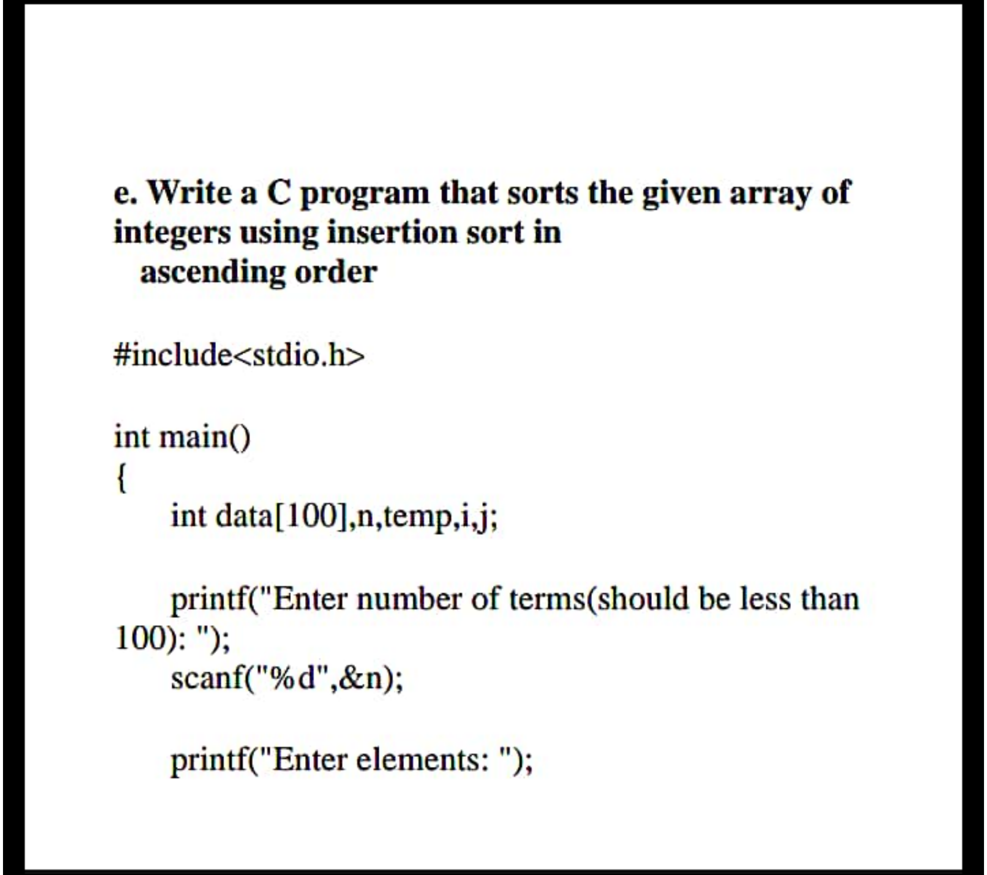
data of array

```
          {
            printf(" %d ", a[i]);
          }

        for (int i = 0; i < n; i++)   //Loop for descending
ordering
        {
          for (int j = 0; j < n; j++)
           {
             if (a[j] < a[i])
              {
               int tmp = a[i];
               a[i] = a[j];
               a[j] = tmp;
              }
            }
        }

      printf("\n\nDescending : ");
      for (int i = 0; i < n; i++)
        {
          printf(" %d ", a[i]);
        }

     return 0;
}
```

RESULT:

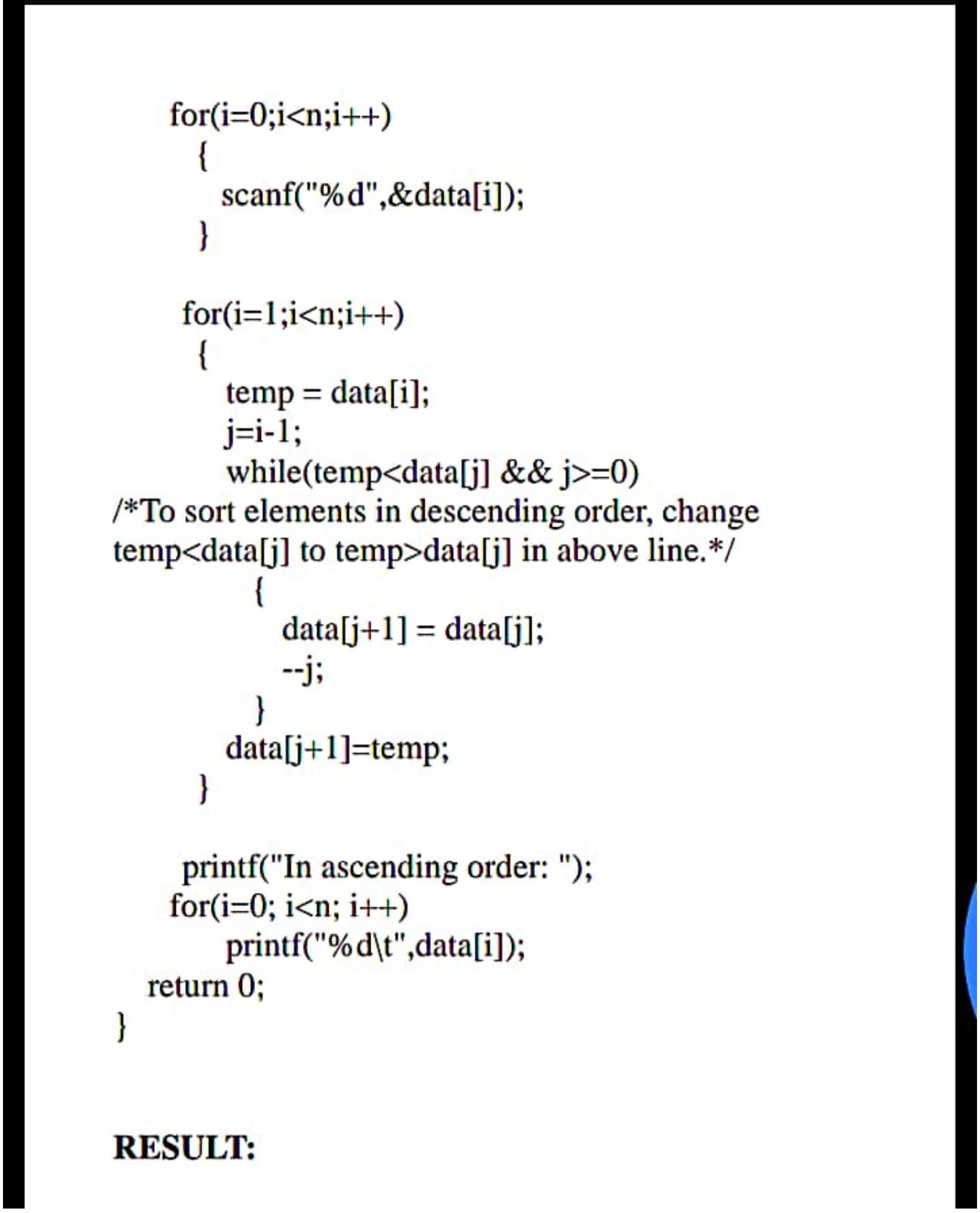
OUTPUT:

```
Array Data :  3  4  7  6  5  1  2  8  10  9

Descending :  10  9  8  7  6  5  4  3  2  1
```



e. Write a C program that sorts the given array of integers using insertion sort in ascending order

```
#include<stdio.h>

int main()
{
     int data[100],n,temp,i,j;

     printf("Enter number of terms(should be less than
100): ");
     scanf("%d",&n);

     printf("Enter elements: ");
```

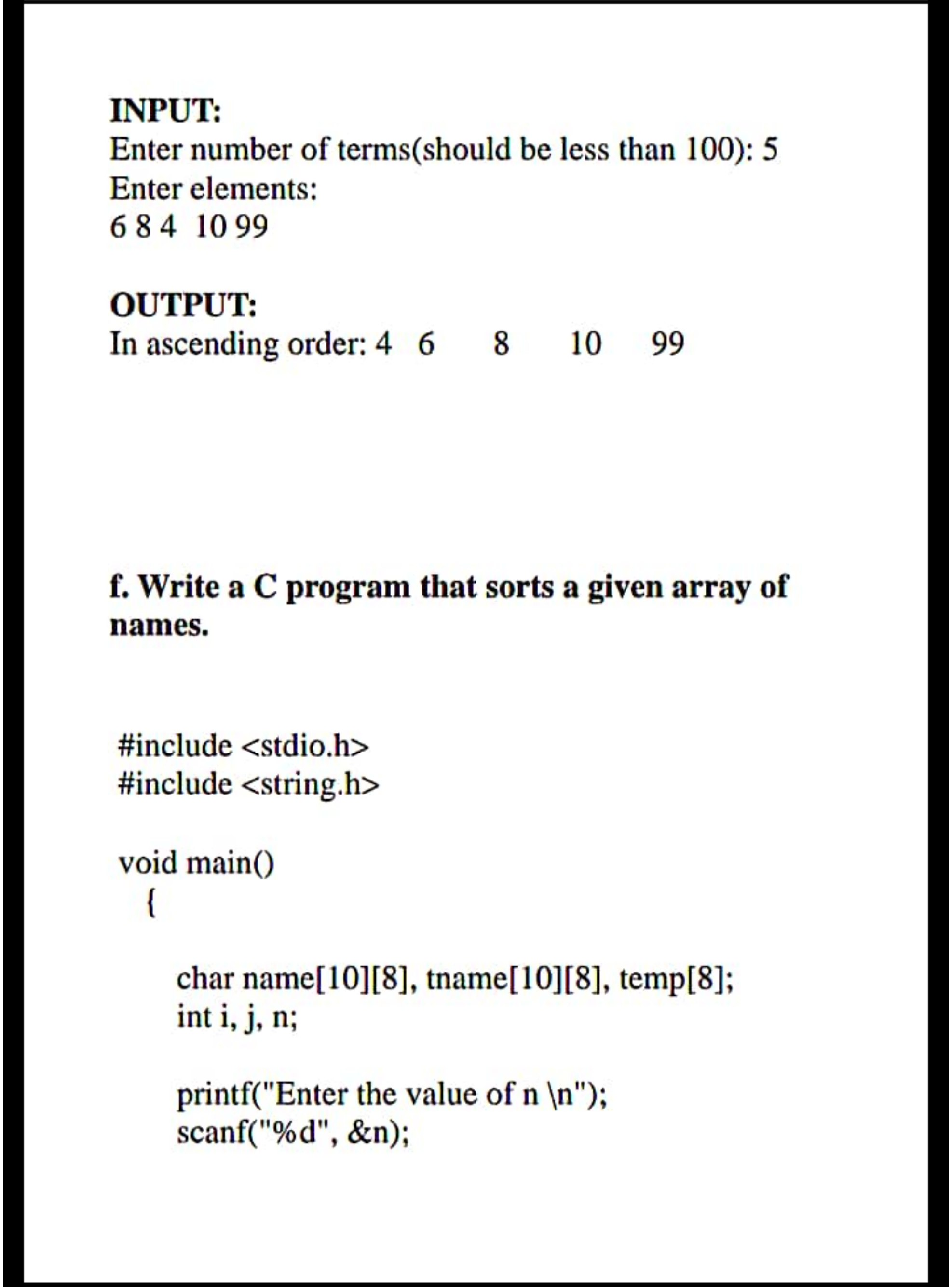


```
     for(i=0;i<n;i++)
       {
        scanf("%d",&data[i]);
       }

      for(i=1;i<n;i++)
       {
         temp = data[i];
         j=i-1;
         while(temp<data[j] && j>=0)
/*To sort elements in descending order, change
temp<data[j] to temp>data[j] in above line.*/
            {
              data[j+1] = data[j];
              --j;
            }
         data[j+1]=temp;
       }

      printf("In ascending order: ");
     for(i=0; i<n; i++)
          printf("%d\t",data[i]);
   return 0;
}
```

RESULT:

```
}
```

RESULT:



INPUT:

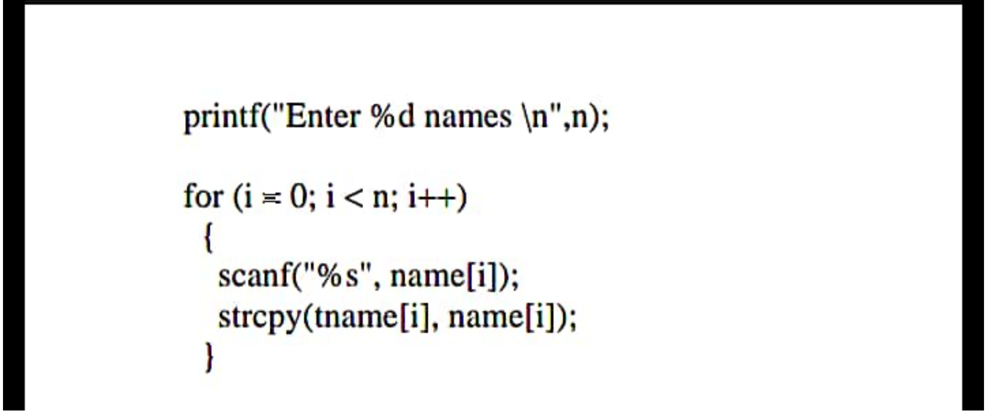
```
Enter number of terms(should be less than 100): 5
Enter elements:
6 8 4  10 99
```

OUTPUT:

```
In ascending order: 4   6      8       10      99
```

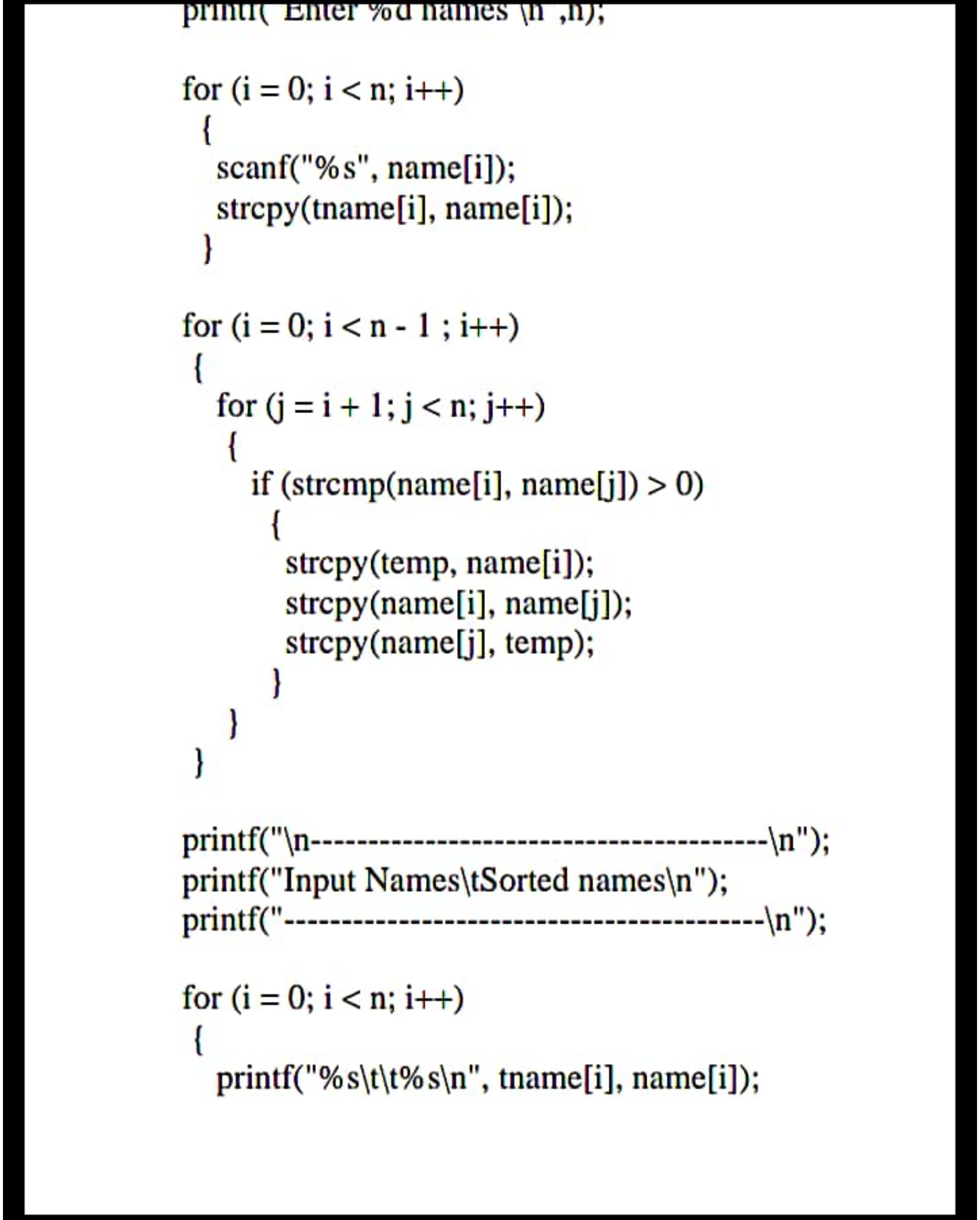
f. Write a C program that sorts a given array of names.

```
#include <stdio.h>
#include <string.h>

void main()
   {

       char name[10][8], tname[10][8], temp[8];
       int i, j, n;

       printf("Enter the value of n \n");
       scanf("%d", &n);
```

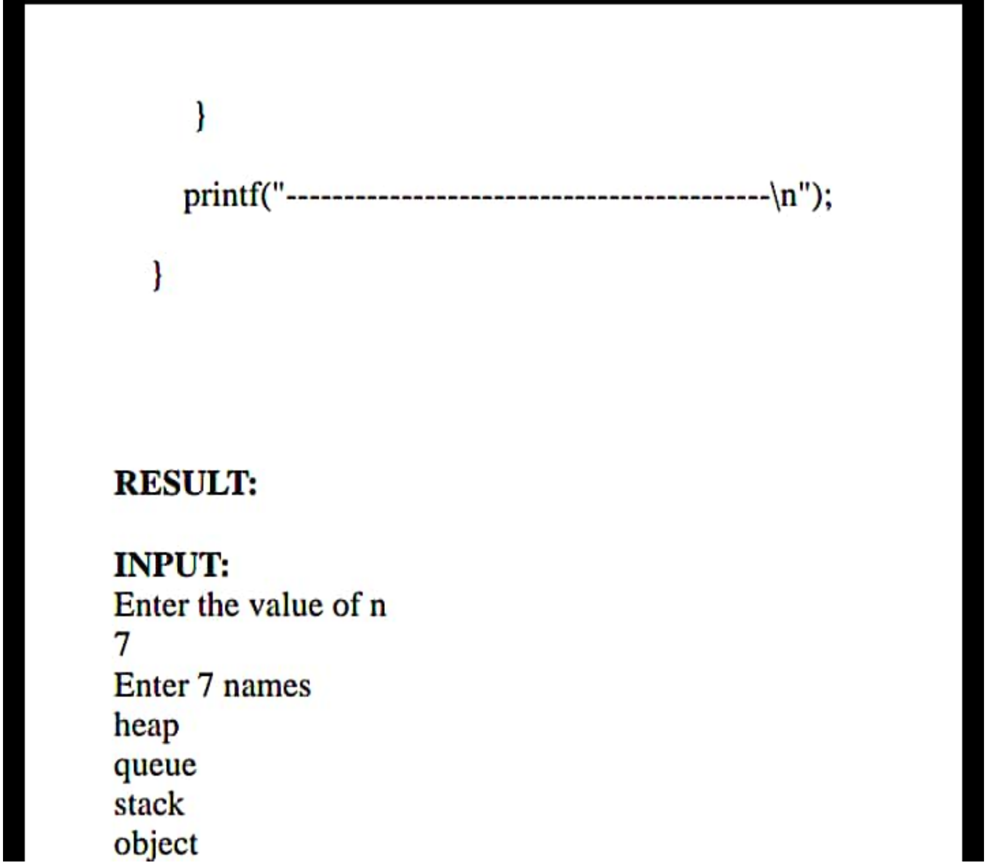


```
       printf("Enter %d names \n",n);

       for (i = 0; i < n; i++)
         {
           scanf("%s", name[i]);
           strcpy(tname[i], name[i]);
         }
```

Scanned by CamScanner



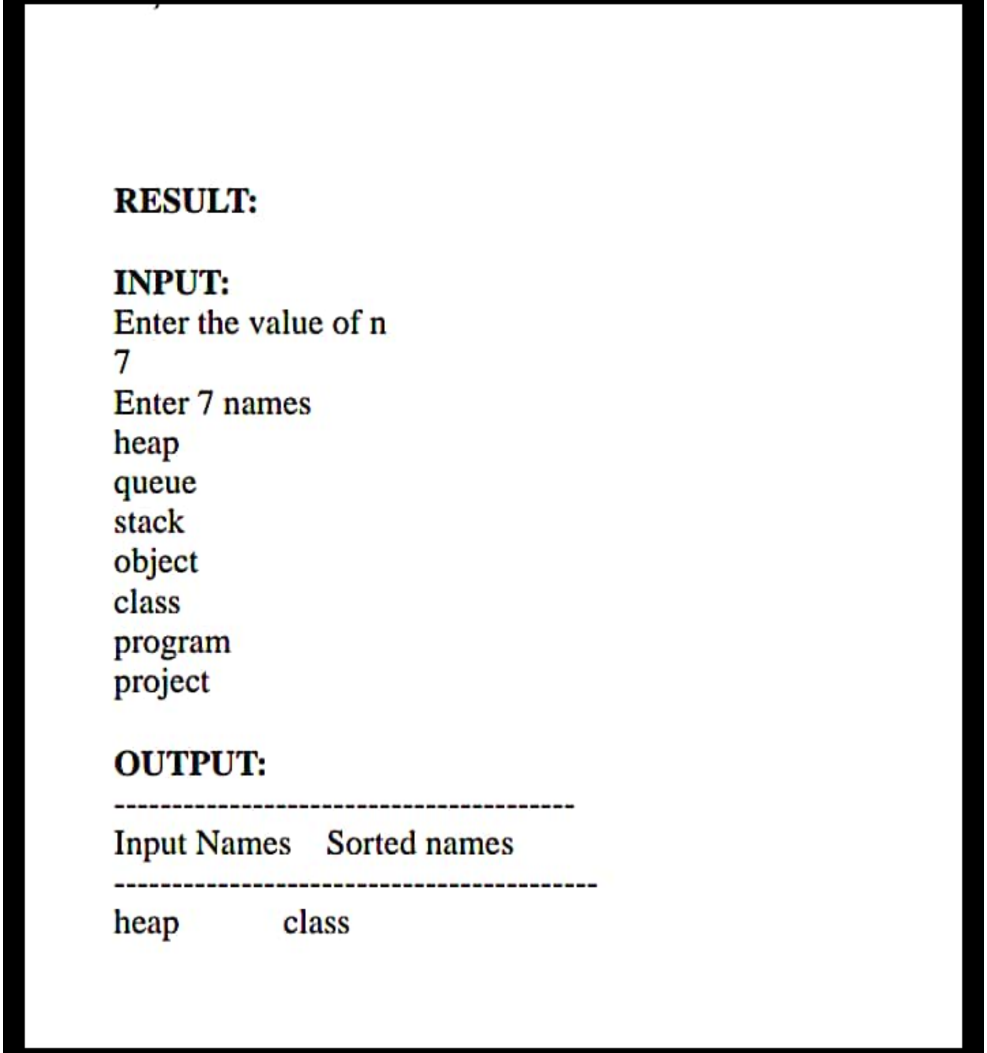
```
    printf("Enter %d names \n",n);

    for (i = 0; i < n; i++)
     {
       scanf("%s", name[i]);
       strcpy(tname[i], name[i]);
     }

    for (i = 0; i < n - 1 ; i++)
    {
       for (j = i + 1; j < n; j++)
        {
          if (strcmp(name[i], name[j]) > 0)
           {
             strcpy(temp, name[i]);
             strcpy(name[i], name[j]);
             strcpy(name[j], temp);
           }
        }
    }

    printf("\n------------------------------------------\n");
    printf("Input Names\tSorted names\n");
    printf("------------------------------------------\n");

    for (i = 0; i < n; i++)
    {
       printf("%s\t\t%s\n", tname[i], name[i]);
```



```
    }

    printf("------------------------------------------\n");

}
```

RESULT:

INPUT:
Enter the value of n
7
Enter 7 names
heap
queue
stack
object



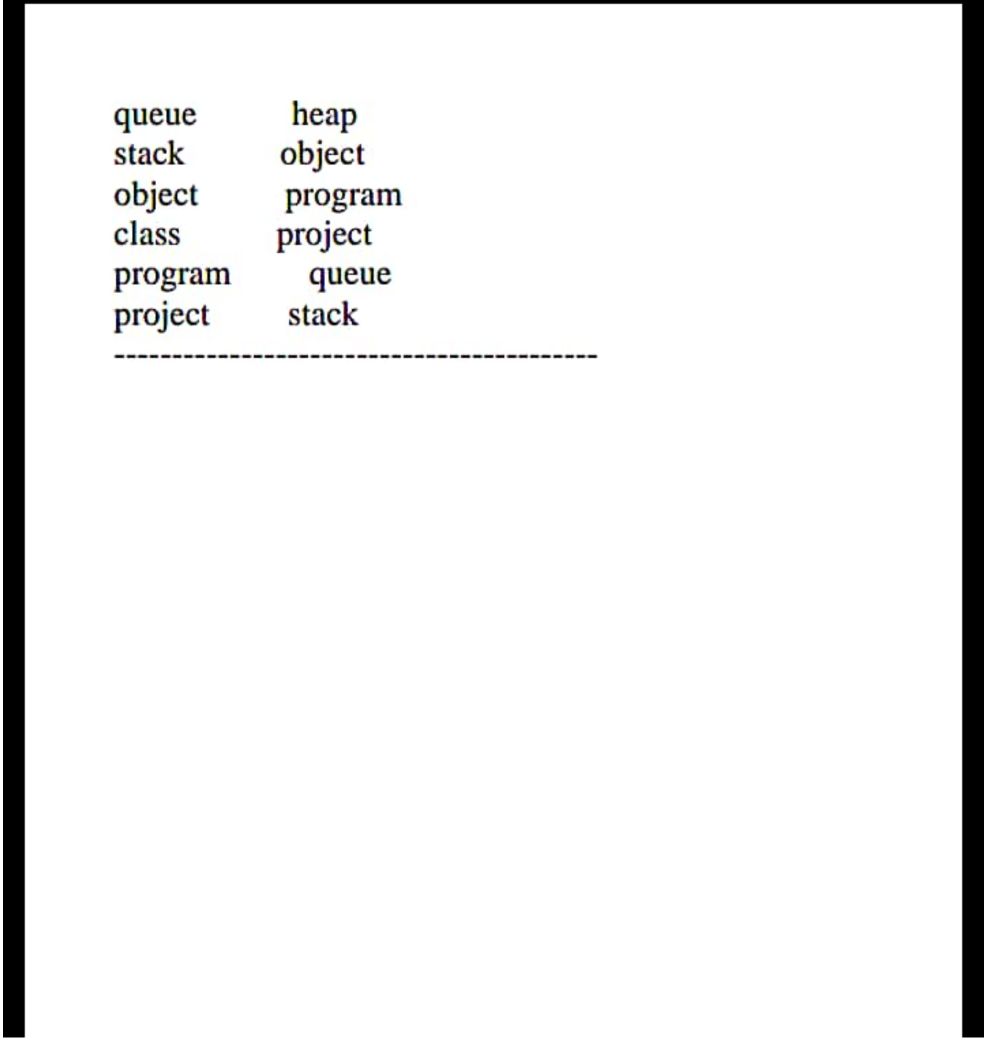
RESULT:

INPUT:

```
Enter the value of n
7
Enter 7 names
heap
queue
stack
object
class
program
project
```

OUTPUT:

```
----------------------------------------
Input Names    Sorted names
------------------------------------------
heap           class
```



```
queue          heap
stack          object
object         program
class          project
program          queue
project        stack
------------------------------------------
```