//Program to Implement Linear & Binary Search - Visakh Bobby - S3R2 - 34

#include <stdio.h>

#include <stdlib.h>

void Lsearch(int A[],int N)

{ printf("Doing Linear Search...\n");

  int item,i;

  printf("Enter the item to be searched\n");

  scanf("%d",&item);

  for(i=0;i<N;i++)

  {

    if(A[i] == item)

    {

      printf("Element %d Found At Position : %d\n",item,i+1);

      break;

    }

  }

  if(i>=N)

    printf("Element Not Found..\n");

}

void Sort(int A[] , int N)

{

  int i,j,temp;

  for(i=0;i<N-1;i++)

  {

    for (j=0;j<N-1-i;j++)

    {

    if(A[j] > A[j+1]) // Ascending Order

      {

        temp = A[j];

        A[j] = A[j+1];

        A[j+1] = temp;

      }

    }

  }

    printf("Sorted Array Is :\n");

    for(i=0;i<N;i++)

      printf("%d\t",A[i]);

    printf("\n");

}

void Bsearch(int A[],int N)

{

  printf("Doing Binary Search...\n");

  int item,n,mid;

  Sort(A,N);

  printf("Enter the item to be searched:\n");

  scanf("%d",&item);

  int lb = 0, ub = N-1;

  while(lb<=ub)

  {

    mid = (lb+ub)/2;

    if(A[mid] == item)

    {

      printf("Element %d Found At Position : %d\n",item,mid+1);

      break;

    }

    else if (A[mid] > item)

      ub = mid -1;

    else // A[mid] < item

      lb = mid +1;

  }

  if(lb>ub) //out of bounds

    printf("Element Not Found\n");

}

void main()

{

  int A[20];

  int N,i,ch=0;

  printf("Enter the number of elements:\n");

  scanf("%d",&N);

  printf("Enter the Elements\n");

  for(i=0;i<N;i++)

    scanf("%d",&A[i]);

  printf("Enter Your Choice:\n");

  printf("1. Linear Search\n");

  printf("2. Binary Search\n");

  printf("3. Sort Array\n");

  printf("4. Exit\n");

  printf("Enter Your Choice:\n");

  scanf("%d",&ch);

  while(ch<=3)

  {

   switch(ch)

   {

     case 1 : Lsearch(A,N);

         break;

     case 2 : Bsearch(A,N);

         break;

     case 3 : Sort(A,N);

         break;

     default : printf("Invalid Choice , Try Again\n");

   }

   printf("Enter Your Choice Again:\n");

   scanf("%d",&ch);

  }

}

**Output:**

