**Program-1:Write a program to search an element using linear search.**
**Objective:**To understand the working of linear search.
**Algorithm:**

```
LinearSearch(arr, n, num)
arr – is an array of n integers
num – is the value to be searched in the array
pos – the position of the value in the array, if found
 1.pos = -1
 2.For x = 0 to n-1 repeat
       If arr[x] == num then
       [ pos = x+1, go to step-3]
 3.If pos = -1 then print "num is not found in the array"
     else print "num is found at position pos"
 4.Stop
```

**Source code:**

```
#include<conio.h>
#include<stdio.h>
void main()
{
 int arr[20],n,pos,x,num;
 printf("Enter the limit of array ");
 scanf("%d",&n);
 printf("Enter the elements in the array ");
  for(x=0;x<n;x++)
   scanf("%d",&arr[x]);
  printf("Enter the number to be searched ");
  scanf("%d",&num);
  pos=-1;
  for(x=0;x<n;x++)
  {
   if(arr[x]==num)
    {
     pos=x+1;
     break;
    }
  }
  if(pos==-1)
   printf("%d not found in an array" );
  else
   printf("%d is found in array at %d position ",num,pos);
 getch();
}
```

**Input 1:**

```
Enter the limit of array 5
Enter the elements in the array 1 2 3 4 5
Enter the number to be searched 3
```

**Output 1:**

```
3 is found in array at 3 position
```

**Input 2:**

```
Enter the limit of array 4
Enter the elements in the array 1 2 3 4
Enter the number to be searched 6
```

**Output 2:**

```
36 not found in an array
```

**Program-2:Write a program to search an element using binary search.**
**Objective:**To understand the working of binary search.
**Algorithm:**

```
Bsearch(arr,n,num)
  arr – is an array of n integers
  num – is the value to be searched in the array
  pos – the position of the value in the array, if found
   1.beg=0,end=n-1,pos = -1
   2.while(beg<=end) repeat
     a.mid=(big+mid)/2
     b.if(num==arr[mid]) then
         {
          pos=mid
          goto step-3
         }
        else if(num<arr[mid]) then end=mimd-1;
        else beg=mid+1
3.if(pos=-1)
    then print"num is not in arr"
    else print"num is in arr at pos "
4.End.
```

**source code:**

```
  #include<conio.h>
  #include<stdio.h>
  void main()
  {
   clrscr();
   int arr[20],lim,beg,num,x,end,mid,pos;
   printf("Enter the limit of array ");
   scanf("%d",&lim);
   printf("Enter the elements in the array ");
    for(x=0;x<lim;x++)
     scanf("%d",&arr[x]);
   printf("Enter the number to be searched ");
   scanf("%d",&num);
   pos=-1;
   beg=0;
   end=lim-1;
   while(beg<=end)
   {
    mid=(beg+end)/2;
    if(num==arr[mid])
     {
      pos=mid+1;
      break;
     }
    else
     if(num<arr[mid])
      end=mid-1;
     else
      beg=mid+1;
   }
   if(pos==-1)
    printf("number is not in array");
   else
    printf("number is in array at position %d",pos);
```

```
   getch();
  }
```

**Input 1:**

```
 Enter the limit of array 4
 Enter the elements in the array 2 5 4 7
 Enter the number to be searched 5
```

**Output1:**

```
 number is in array at position 2
```

**Input 2:**

```
 Enter the limit of array 4
 Enter the elements in the array 2 5 4 7
 Enter the number to be searched 3
```

**Output 2:**

```
 number is not in array
```

**Program-3:** Write a program to Insert an element from an array
**Objective:** To understand the Insertion of an element from an array
**Algorithm:**

```
Insert (A,n,x,pos)
let a is an array of size
n is the number of elements in the array
x is the data to be inserted in the array
Step 1: IF (N=size) then
    [ Print "Array Full"]
Step 2: FOR i = n to pos, step -1, repeat
    [ A[i]= A[i-1]
Step 3: A[Pos - 1] = x
Step 4: End.
```

**Source Code:**

```
#include <stdio.h>
#include<conio.h>
int main()
{
int array[100], position, c, n, value;
printf("Enter number of elements in array\n”);
scanf("%d", &n);
printf("Enter %d elements\n", n);
for (c = 0; c < n; c++)
scanf("%d", &array[c]);
printf("Enter the location where you wish to insert an element\n”);
scanf("%d", &position);
printf("Enter the value to insert\n”);
scanf("%d", &value);
for (c = n - 1; c >= position - 1; c—)
array[c+1] = array[c];
array[position-1] = value;
printf("Resultant array is\n");
for (c = 0; c <= n; c++)
 printf("%d\n", array[c]);
 return 0;
}
```

**Output:**

```
Enter number of elements in array
5
Enter 5 elements
2
3
4
5
6
Enter the location where you wish to insert an element
3
Enter the value to insert
10
Result array is
2
3
10
4
5
6
```

**Program-4:** Write a program to Delete an element from an array
**Objective:** To understand the Deletion of an element from an array
**Algorithm:**
1. Start
2. Set J = K
3. Repeat steps 4 and 5 while J < N
4. Set A[J] = A[J + 1]
5. Set J = J+1
6. Set N = N-1
7. Stop

**Source Code:**

```
#include<stdio.h>
#include<conio.h>
main()
{
int A[] = {1,3,5,7,8};
int k = 3, n = 5;
int i, j;
printf("The original array elements are :\n");
for(i = 0; i<n; i++) {
printf("A[%d] = %d \n", i, A[i]);
}
j = k;
while( j < n) {
A[j-1] = A[j];
j = j + 1;
}
n = n -1;
printf("The array elements after deletion :\n");
for(i = 0; i<n; i++) {
printf("A[%d] = %d \n", i, A[i]);
}
return 0;
}
```

**Output:**

```
The original array elements are:
A [0] = 1
A [1] = 3
A [2] = 5
A [3] = 7
A [4] = 8
The array elements after deletion:
A [0] = 1
A [1] = 3
A [2] = 7
A [3] = 8
```

**Program-5:** Write a program to Sort elements in the array
**Objective:** To understand the Sorting of elements.
**Algorithm:**

```
Sort (A,n)
FOR i=0 to n-2 repeat
[ MIN = A[i], Pos = i
FOR J=i+1 to n-1 repeat
[ IF (A(J) < MIN )
[ MIN = A(J)
Pos = J
]
]
Temp = A[i]
A[i] = Pos
A[Pos] = Temp
]
End
```

**Source Code:**

```
#include<stdio.h>
#include<conio.h>
int main()
{
int array[100], n, c, d, position, swap;
printf("Enter number of elements\n”);
scanf("%d", &n);
printf("Enter %d integers\n", n);
for ( c = 0 ; c < n ; c++ )
scanf("%d", &array[c]);
for ( c = 0 ; c < ( n - 1 ) ; c++ )
{
position = c;
for ( d = c + 1 ; d < n ; d++ )
{
if ( array[position] > array[d] )
position = d;
}
if ( position != c )
{
swap = array[c];
array[c] = array[position];
array[position] = swap;
}
}
printf("Sorted list in ascending order:\n”);
for ( c = 0 ; c < n ; c++ )
printf("%d\n", array[c]);
return 0;
}
```

**Output:**

```
Enter number of elements in array
10
Enter 10 elements
6
2
10
9
```

```
5
1
-2
8
12
45
Sorted list in ascending order
-2
1
2
5
6
8
9
10
12
45
```