

1. Binary Search (Iterative)

Algorithm

1. start
2. accept sorted list of number arr and length of list , len
3. accept element e to be searched for in the list
4. set beg=0,end=len
5. repeat while beg <= len
 1. set mid = (beg + end)/2
 2. if arr[mid]= e , then print found at mid , break
 3. else if arr[mid] > e , then set end = mid - 1
 4. else , set beg = mid + 1
6. if beg <= len , print not found
7. stop

Sorting algorithm (Bubble Sort)

1. start
2. accept list of numbers , arr and length of list , len
3. set i = 0
4. repeat while i < len
 1. set j=0
 2. repeat while j < len - i - 1
 1. if arr[j] > arr[j+1] , then swap arr[i] and arr[j+1]
 2. j = j + 1
 3. i = i + 1
5. stop

Source Code

```
#include <stdio.h>

int main()
{
    int x, y, flag = 0;
    printf("\n To search an element from 2D array : \n");

    printf("\n Enter n.o rows in array : ");
    scanf("%d", &x);

    printf("\n Enter n.o cols in array : ");
    scanf("%d", &y);

    int arr[x][y], i, j, search;

    for (i = 0; i < x; i++) {
        printf("enter row %d : \n", i + 1);
        for (j = 0; j < y; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
}
```

```

printf("\n Enter element to be searched : ");
scanf("%d", &search);

for (i = 0; i < x; i++)
{
    for (j = 0; j < y; j++)
    {
        if (arr[i][j] == search)
        {
            flag = 1;
            printf("\n\n %d Found at position (%d,%d) ", search, i + 1, j + 1);
        }
    }
}
if (!flag)
    printf("\n\n Not found ");
return 0;
}

```