Rajshahi University of Engineering & Technology
Department of Computer Science of Engineering

EXPERIMENT NO: 02
NAME OF EXPERIMENT: Arrays, Records and Pointers

SUBMITTED TO:

RIZOAN TOUFIQ
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

SUBMITTED BY:

NAME: MD. ARIFUL ISLAM
ROLL NO.: 1803046
GROUP: 2ND THIRTY
DATE OF EXP.: 08-09-2019
DATE OF SUB. : 16-09-2019
SERIES: 18 SERIES

MACHINE CONFIGURATION:
ASUS X510UF
CORE I5 8TH GEN PROCESSOR
UP TO 3.4 GHZ
8 GB RAM
OS WIN 10

INDEX

THEORY: Data structures are classified as either Linear or Nonlinear. A data structure is said to be linear if its elements are from a sequence or from a linear list. There are two basic ways to represrnting a linear structure in memory:

1. **Array.**
2. **Linked list.**

The operations one normally performs on any linear structure are include the following :

1. **Traversal**
2. **Search**
3. **Inserting**
4. **Deletion**
5. **Sorting**
6. **Merging Etc.**

This Lab is just about Array. In next Labs we will work with Linked lists.

A **Linear Array** is a list of finite number **n** of **Homogeneous** data elements (i.e., data elements of the same type) such that:

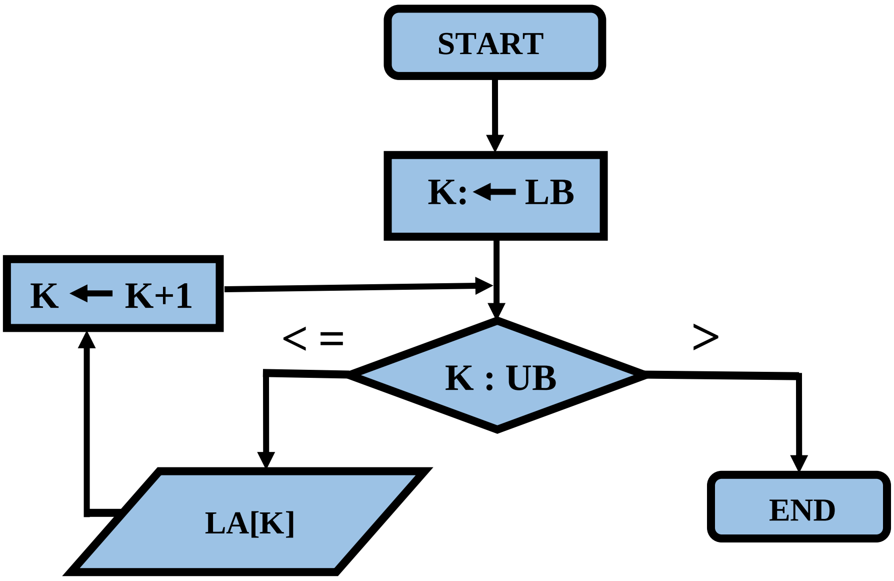
1. The elements of the array are referenced respectively by an Index set consisting of n consecutive numbers.

2. The elements of the **Array** are stored respectively in successive memory locations.

The number **n** of elements is called the **Length** or **Size** of the Array**.**

PROBLEM 1: TRAVERSING LINEAR ARRAY.

FLOW CHART:



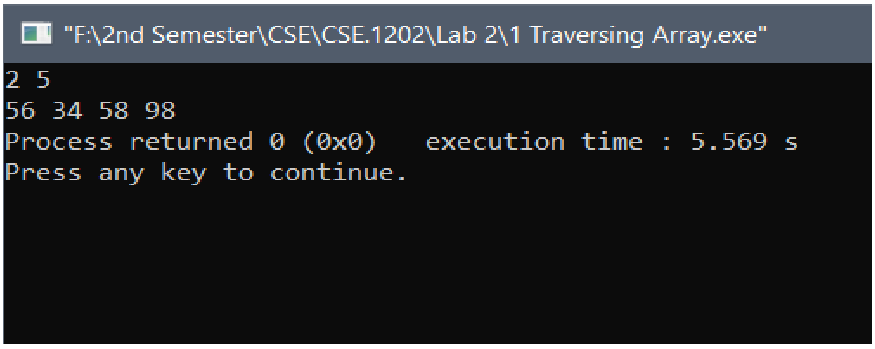
ALGORITHM:

```
(Given a linear array LA with Lower bound LB and upper bound UB.
This algorithm traverses LA)
1. Set K := LB
2. Repeat 3 and 4 while K≤UB
3. Write: LA[K].
4. Set K:=K+1.
        [End of step 2 loop]
5. Exit
```

CODE:

```
#include<stdio.h>

int main()
{
   int K,LB,UB,LA[10]={23,56,34,58,98,86,75,71,91,47};

   scanf("%d %d",&LB,&UB);

   K=LB-1;
   while(K<UB)
   {
      printf("%d ",LA[K]);
      K=K+1;
   }

   return 0;
}
```

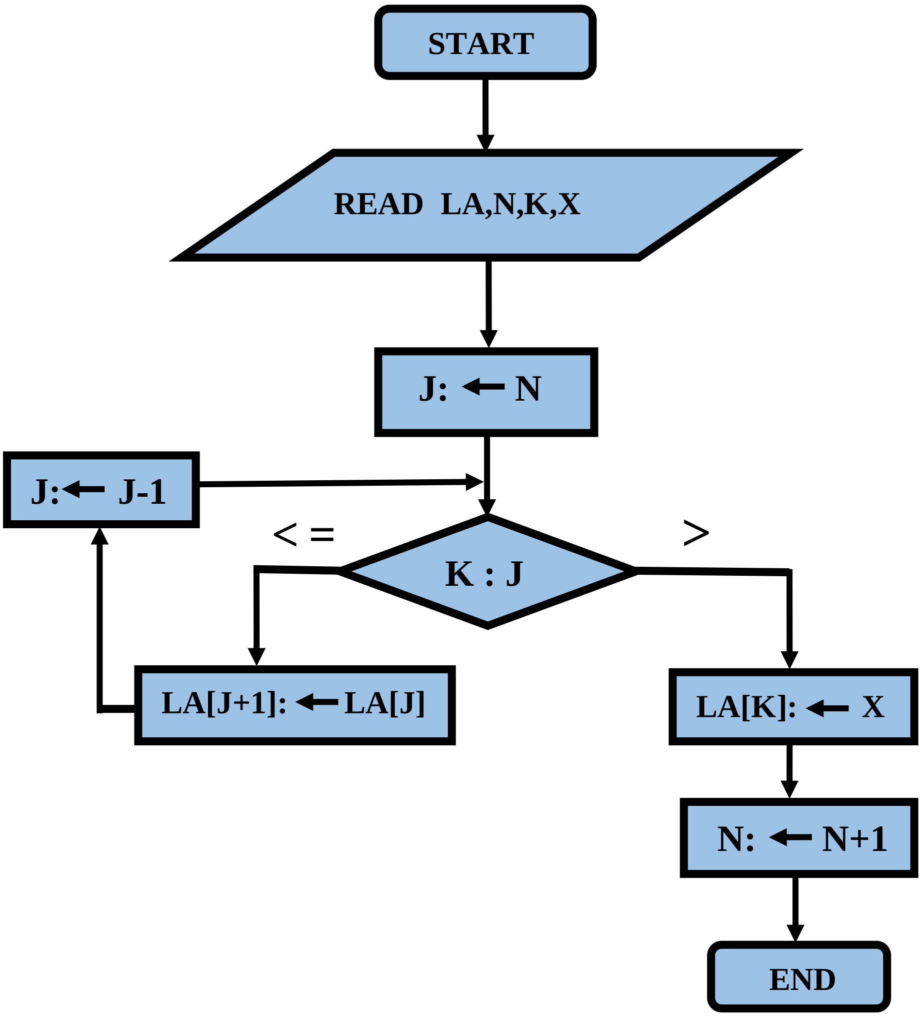
OUTPUT:



```
2 5
56 34 58 98
Process returned 0 (0x0)   execution time : 5.569 s
Press any key to continue.
```

PROBLEM 2: INSERTING INTO A LINEAR ARRAY.

FLOW CHART:



ALGORITHM:

INSERT (LA, N, K, X)

(Here LA is a linear array with N elements and K is a positive integer such that K≤N. This algorithm inserts an element x into the Kth position in LA)

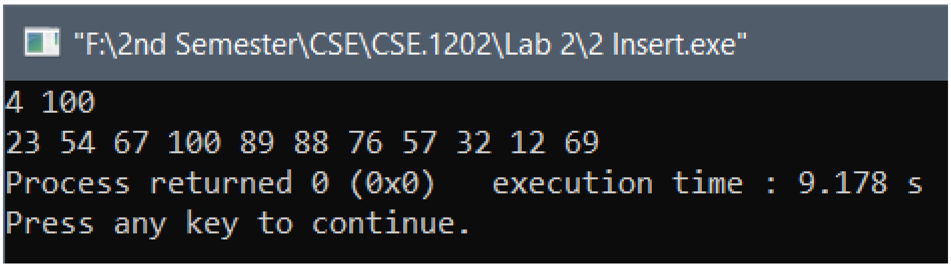
```
1. Set J:=N.
2. Repeat steps 3 and 4 while K≤J
3. Set LA[J+1] := LA[J]
4. J:=J-1.
```

```
        [End of step 2 loop]
    5. Set LA[K]:=X.
    6. Set N:=N+1.
    7. Exit
```

CODE:

```
#include<stdio.h>

int main()
{
    int J,K,X,LA[100]={23,54,67,89,88,76,57,32,12,69},N=10;

    scanf("%d %d",&K,&X);

    K=K-1;
    J=N-1;
    while(K<=J)
    {
        LA[J+1]=LA[J];
        J=J-1;
    }

    LA[K]=X;
    N=N+1;
    for(J=0;J<N;J++)
        printf("%d ",LA[J]);

    return 0;
}
```

OUTPUT:

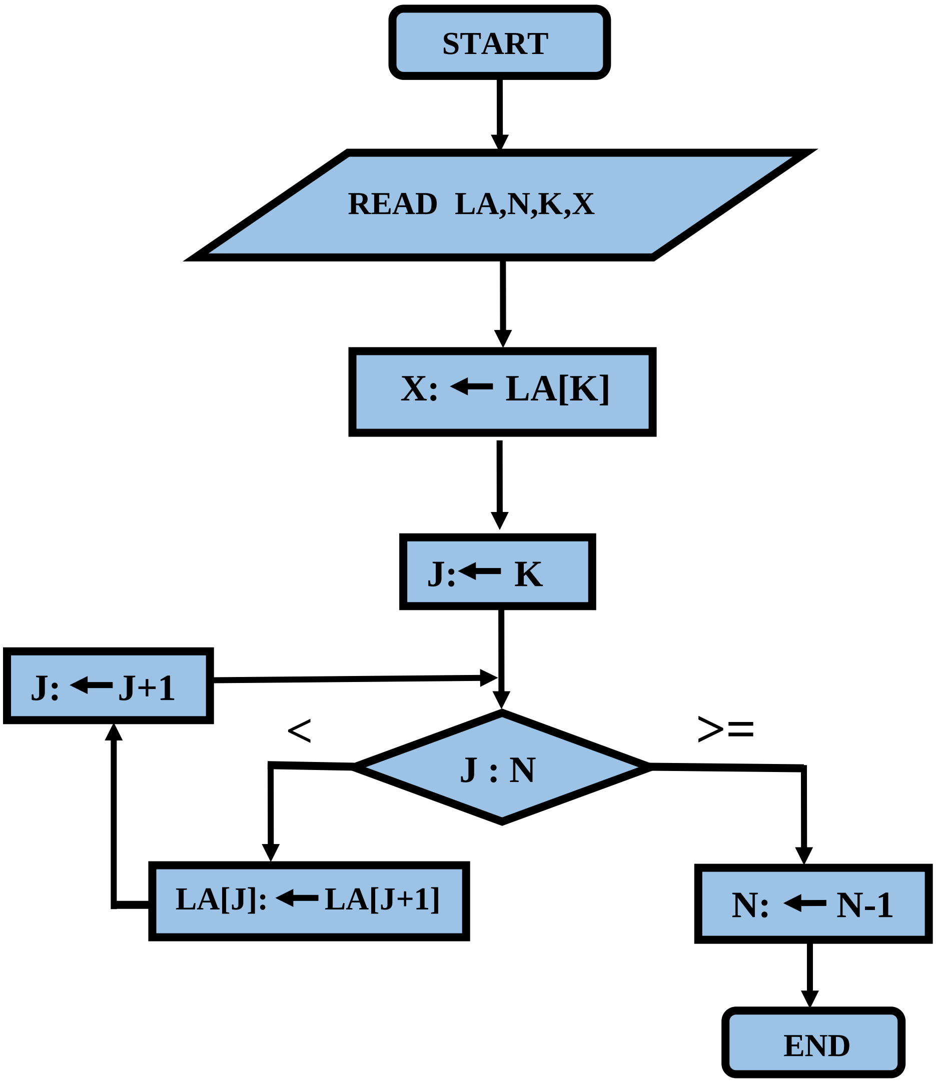


"F:\2nd Semester\CSE\CSE.1202\Lab 2\2 Insert.exe"

```
4 100
23 54 67 100 89 88 76 57 32 12 69
Process returned 0 (0x0)   execution time : 9.178 s
Press any key to continue.
```

PROBLEM 3: DELETING FROM A LINEAR ARRAY.

FLOW CHART:



ALGORITHM:

INSERT (LA, N, K, x) (Here LA is a linear array with N elements and K is a positive integer such that K≤N. This algorithm deletes the Kth element from LA)

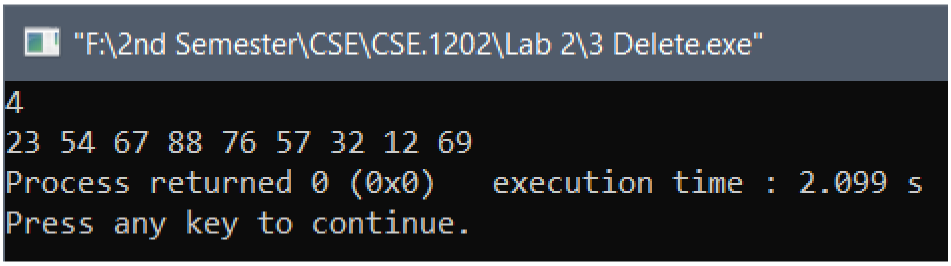
1. Set x:=LA[K] and J :=K.

```
2. Repeat steps 3 and 4 while K<N
3. Set LA[J] := LA[J+1]
4. J:=J+1.
        [End of step 2 loop]
5. Set N:=N-1.
6. Exit
```

CODE:

```
#include<stdio.h>

int main()
{
   int J,K,X,LA[100]={23,54,67,89,88,76,57,32,12,69},N=10;

   scanf("%d",&K);

   X=LA[K-1];
   J=K-1;
   while(J<N-1)
   {
      LA[J]=LA[J+1];
      J=J+1;
   }

   N=N-1;
   for(J=0;J<N;J++)
      printf("%d ",LA[J]);


   return 0;
}
```

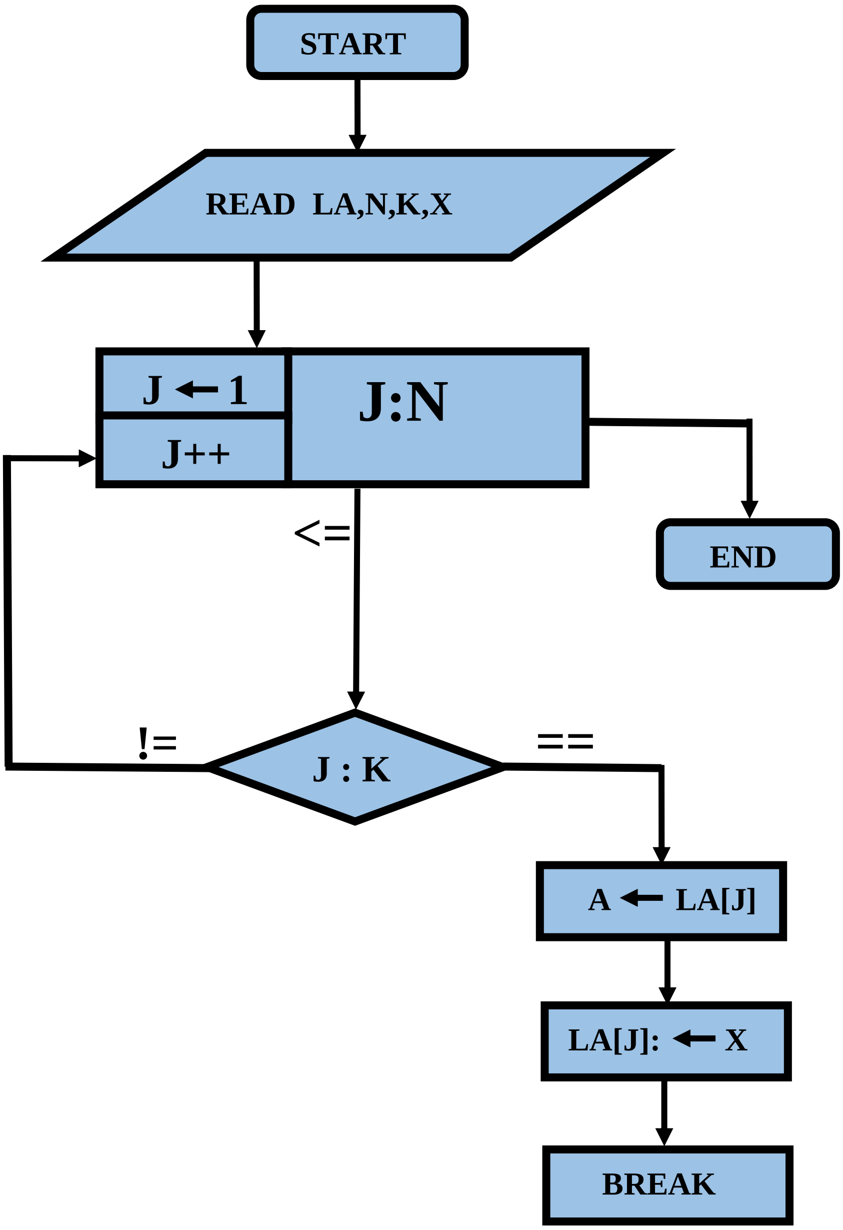
OUTPUT:



```
4
23 54 67 88 76 57 32 12 69
Process returned 0 (0x0)   execution time : 2.099 s
Press any key to continue.
```

PROBLEM 4: UPDATE THE VALUE OF KTH VALUE OF A LINEAR ARRAY.

FLOW CHART:



ALGORITHM:

INSERT (LA, N, K, X) (Here LA is a linear array with N elements and K is a positive integer such that K≤N. This algorithm updates the Kth element of LA)

1. Repeat for J = 1 to N by 1

```
2. If J = K then:
A = LA[J].
LA[J] = X.
[End of step 1 loop].
3. Exit
```

CODE:

```
#include<stdio.h>

int main()
{
   int A,X,I,K;
   int LA[10]={34,67,87,54,26,84,59,73,89,97},N=10;

   scanf("%d %d",&K,&X);
   for(I=0;I<N;I++)
      {
         if(I==(K-1))
          {
             A=LA[I];
             LA[I]=X;
          }
      }
   for(I=0;I<N;I++)
      printf("%d  ",LA[I]);

   return 0;
}
```

DISCUSSION: The problems were based on Array. After trying sometime I had solved the problems.

The End