```
/* ARRAY [definition, operations (insert and delete, traverse)
// Applications (searhing sorting)
// relation with pointers  => string (character array) */

//define array   int a=5;  2Bytes [-32768 to +32767]

// int a=5, b=6, c=7
/*
  variable name  value    address (let)
    a          5       100
    b          6        58
    c          7       250
*/
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/* int a[5];               // a var =>  a[] indicates it is an array    => a[5] a is an
array with 5 cells

index      0   1   2   3   4
value      ?   ?   ?   ?   ?
m. addr     10  12  14  16  18

float a[5];

index      0   1   2   3   4
value      ?   ?   ?   ?   ?
m. addr     20  24  28  32  36

int a[10] = {23,34,56,78,12};
*/



#include <stdio.h>
int main()
{
   int a[100], n, i, data, pos;
```

```
  do{
  printf("\nEnter the no of inputs (within 0 to 99)");
  scanf("%d",&n);          // n will b d array size   n=5
  }while(n<0 || n>99);   //

  for(i=0;i<n;i++)                    // scanning inputs
  {
     printf("\nenter the value for a[%d]",i);
     scanf("%d",&a[i]);
  }

  //_____________________________________________________
  printf("\nArray looks like below...");     // printing array values
  for(i=0;i<n;i++)
    printf("%5d\t", a[i]);

   //_____________________________________________________
   // Now we will insert some element to an array....
   // Inserting an element into an array

   //scanning an element
   printf("\nEnter the data to be insert... ");
   scanf("%d",&data);

   //scanning for position
   do{
      printf("\nEnter the position... ");
      scanf("%d",&pos);
   }while(pos<0 || pos>n);       // array size 6 (0-5) <0      // }while(pos<0 ||
pos>n);

   // 0 1 2 3 4 5 6    // array indices [positions]
   // 1 2 3 4 5 6    // array size 6 n=6    pos=-1 ?? NO   pos<n   // pos 2  and
data 9  (let)
   // 1 2 3 3 4 5 6  (insertion done)
```

```
    i=n;   // size of an array
    while(i>pos)                    // SHIFTER
    {
        a[i]=a[i-1];        // a[6]= a[5]   a[5]= a[4]  a[4] =a[3]   a[i] = a[i-1]
 CLEAR ????
        i--;
    }                       // 1 2 3 3 4 5 6  (insertion not yet)

    a[pos] = data;     // insert the element into that position
                           // 1 2 9 3 4 5 6  (insertion DONE)
//_____________________________________________________
   printf("\nArray looks like below...");     // printing array values
   for(i=0;i<n+1;i++)
      printf("%5d\t", a[i]);
   return 0;
}
```

ARRAY DELETION

```
/* ARRAY [definition, operations (insert and delete, traverse)

// Applications (searching sorting)


// relation with pointers  => string (character array) */

//define array   int a=5;  2Bytes [-32768 to +32767]

// int a=5, b=6, c=7
/*
  variable name  value    address (let)
    a          5       100
    b          6        58
    c          7       250
*/
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/* int a[5];                // a var =>  a[] indicates it is an array    => a[5] a is an
array with 5 cells

index      0   1   2   3   4
value      ?   ?   ?   ?   ?
m. addr     10  12  14  16  18

float a[5];

index      0   1   2   3   4
value      ?   ?   ?   ?   ?
m. addr     20  24  28  32  36

int a[10] = {23,34,56,78,12};
*/
```

```
#include <stdio.h>
int main()
{

  int a[100]={23, 34, 56, 78}, n=4, i, pos=1;

/*   do{


  printf("\nEnter the no of inputs (within 0 to 99)");
  scanf("%d",&n);          // n will b d array size   n=5
  }while(n<0 || n>99);   //


  for(i=0;i<n;i++)                    // scanning inputs
  {
     printf("\nenter the value for a[%d]",i);
     scanf("%d",&a[i]);
  }

*/
  //_____________________________________________________
  printf("\nArray looks like below...");     // printing array values (traverse)


  for(i=0;i<n;i++)
    printf("%5d\t", a[i]);

   //_____________________________________________________

   // Delete an element from some given position

   //scanning for position
 /*
```

```
    do{
        printf("\nEnter the position... ");
        scanf("%d",&pos);
    }while(pos<0 || pos>n);       // array size 6 (0-5) <0      // }while(pos<0 ||
pos>n);

*/

    // 0    1   2   3     index values
    // 23, 34, 56, 78     array elements    // pos =1    n=4

    data= a[pos];

    i=pos;
    while(i<n-1)                    // LEFT SHIFTER
    {
        a[i]=a[i+1];        // a[1] = a[2]    a[2]= a[3]  a[3]=a[4] ....
        i++;
    }
    n--;     //array size will be reduced by 1

//_____________________________________________________
   printf("\nArray looks like below...");     // printing array values
   for(i=0;i<n;i++)
      printf("%5d\t", a[i]);

   return 0;
}
```

LINEAR SEARCHING

```
// linear searching => modified linear searching

#include <stdio.h>
int main()
{
  int a[100]={23, 34, 56, 78, 112}, n=5, i, data= 56;


/*   do{
  printf("\nEnter the no of inputs (within 0 to 99)");
  scanf("%d",&n);          // n will b d array size   n=5
  }while(n<0 || n>99);   //


  for(i=0;i<n;i++)                    // scanning inputs
  {
     printf("\nenter the value for a[%d]",i);
     scanf("%d",&a[i]);
  }
 */
  //_____________________________________________________
  printf("\nArray looks like below...");     // printing array values (traverse)
  for(i=0;i<n;i++)
    printf("%5d\t", a[i]);


   //_____________________________________________  LINEAR
SEARCHING
  for(i=0;i<n;i++)
  {
    if(a[i] == data)
      printf("\n\n%5d is found at %3dth Location\t", a[i], i);
```

```
    }
    return 0;
}

// problems of linear searching:  2 times checking
```

MODIFIED LINEAR SEARCHING

```
// Modified linear searching

#include <stdio.h>
int main()
{
  int a[100]={0, 23, 34, 56, 78, 112}, n=5, i, data= 300;


/*   do{
  printf("\nEnter the no of inputs (within 0 to 99)");
  scanf("%d",&n);          // n will b d array size   n=5
  }while(n<0 || n>99);   //


  for(i=0;i<n;i++)                    // scanning inputs
  {
     printf("\nenter the value for a[%d]",i);
     scanf("%d",&a[i]);
  }
 */

    // Modified linear SEARCHING
    a[0]= data;
    i= n;
    while(a[i]!=data)
       i--;


   printf("\n\n%5d is found at %3dth Location\t", a[i], i);
  return 0;
}
```

```
// index   0   1   2   3   4   5
// array   300, 23, 34, 56, 78, 112              i=5
```