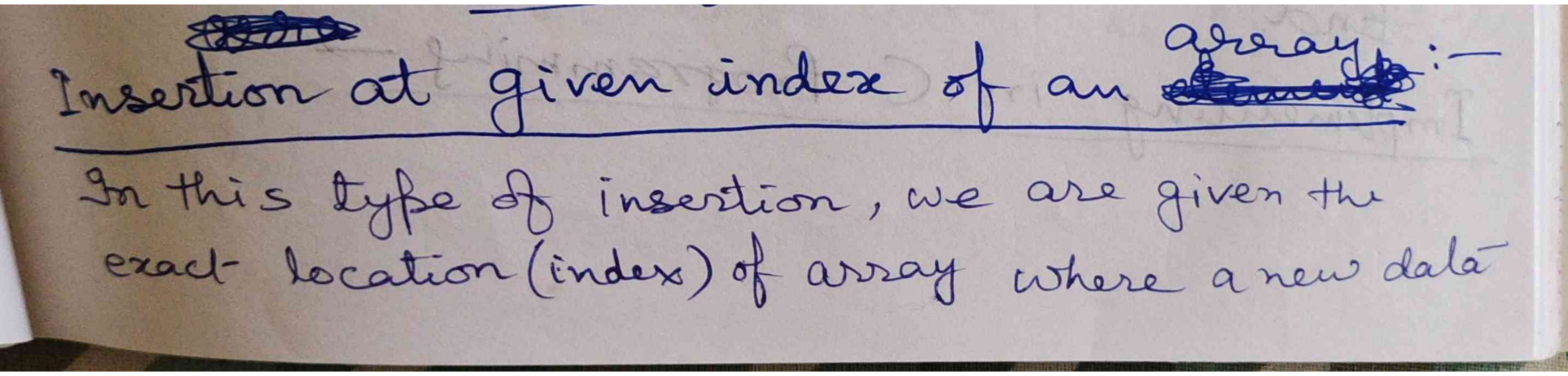

Insertion at given index of an array :-

In this type of insertion, we are given the exact location (index) of array where a new data element needs to be inserted.

First, we shall check if the array is full; if it is not, then we shall move all data elements from that location one step downward. This will make room for new data element.

Algorithm — Let A is an array with N elements. The maximum numbers. of element it can store is defined by MAX.

```
Begin
If N = MAX, return
ELSE
     N = N + 1
     SEEK Location index
     For All Elements from A [index] to
     A[N]
       Move to next adjacent location
       A[index] = new_element
End
```

Implementing in C Programming →

```
#include <stdio.h>
 #define  MAX  5
 void main(){
      int  array[MAX] = {11, 12 , 14 , 15};
     int  N=4; // number of elements in array
     int  i=0; //loop variable
     int index =2; //index location to insert new value
     int  value =13; // new data element to be inserted
  printf("Printing array before insertion:\n");
  for (i=0 ;  i< N ;  i++) {
   printf ("array[%d] = %d \n", i, array[i]);
   }
 for (i= N ;   i >= index ;  i--) {
      array[i + 1] = array[i];
     }
  array[index] = value;
   N++;
 printf(" Printing array after insertion :\n");
  for(i=0; i<N ; i++){
  printf (" array[%d] =  %d \n", i, array[i]);
  }
getch();
}
```

The program should yield the following result —

```
Printing array before insertion :
   array [0] = 11
   array [1] = 12
   array [2] = 14
   array [3] = 15
Printing array after insertion :
  array [0] = 11
  array [1] = 12
  array [2] = 13
  array [3] = 14
  array [4] = 15
```