

## Insertion at the beginning of an array:-

When we insert at beginning, then every existing data elements to shift one step downward.

Algorithm:- Let us assume  $A$  is an array with  $N$  elements. The maximum number of elements it can store is defined by  $MAX$ .

We shall first check if array has any empty space to store an element and then we proceed with insertion operation (process).

Begin

If  $N = \text{MAX}$ , return

ELSE

$N = N + 1$

For All Elements in A

Move to next adjacent-location

$A[\text{FIRST}] = \text{new\_element}$

End

Implementation in C programming :-

```
#include <stdio.h>
```

```
#define MAX 5
```

```
void main() {
```

```
    int array[MAX] = {12, 13, 14, 15};
```

```
    int N = 4; // Number of elements in array
```

```
    int i = 0; // loop variable
```

```
    int value = 10; // new data to be stored
```

```
    printf("Printing array before insert:\n");
```

```
    for (i = 0; i < N; i++) {
```

```
        printf("array[%d] = %d\n", i, array[i]);
```

```
    }
```

```
    for (i = N; i >= 0; i--) {
```

```
        array[i+1] = array[i];
```

```
    }
```

```

array[0] = value;
N++;
printf("Printing array after insert : \n");
for (i = 0; i < N; i++) {
    printf("array [%d] = %d \n", i, array[i]);
}
getch();
}

```

Output of the above program :-

Printing array before insert :

array [0] = 12

array [1] = 13

array [2] = 14

array [3] = 15

Printing array after insert :

array [0] = 10

array [1] = 12

array [2] = 13

array [3] = 14

array [4] = 15