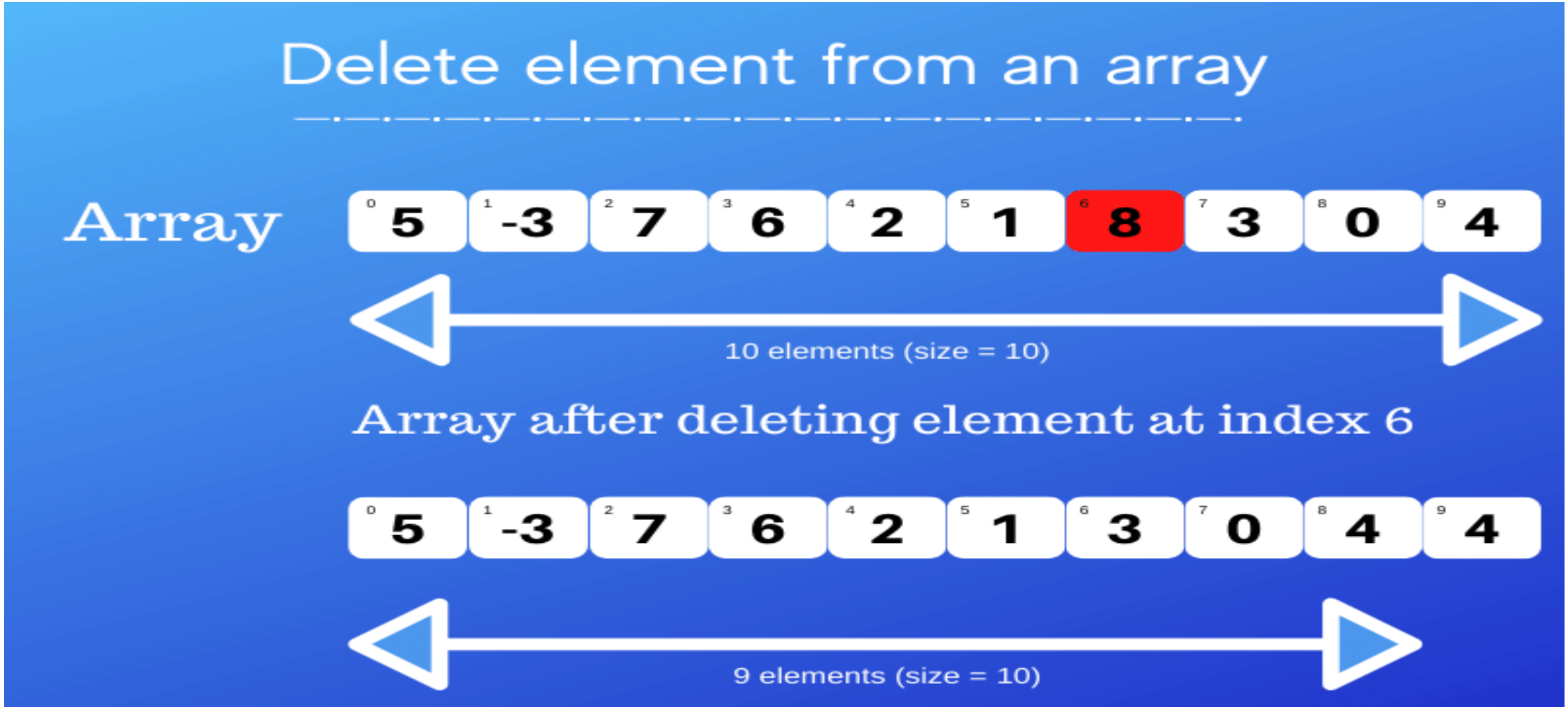
Arrays

Deletion

C program to delete an element from an array



Given an array and a number ‘x’, write a program to delete ‘x’ from the given array.

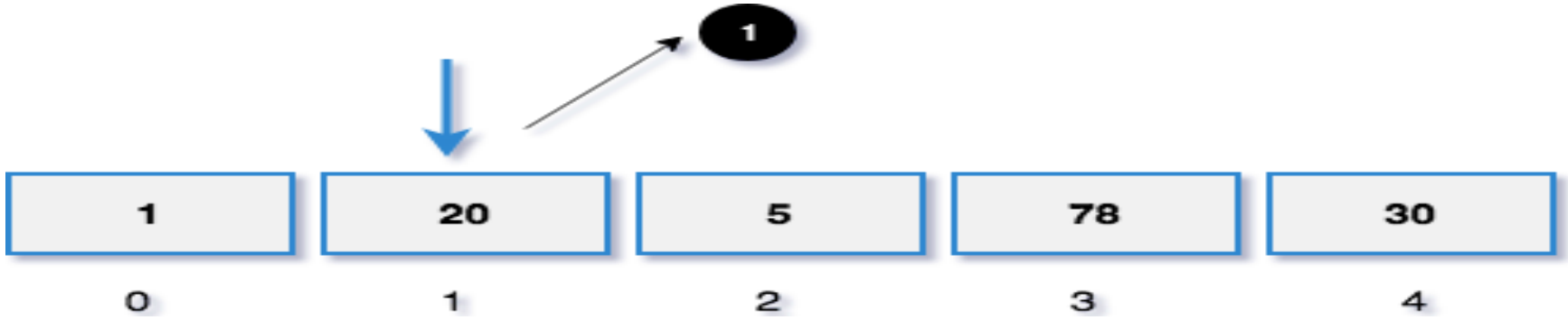
- We assume that array maintains two things with it,
  - capacity and
  - size.
- So when we remove an item, capacity does not change, only size changes.
- **Example:**

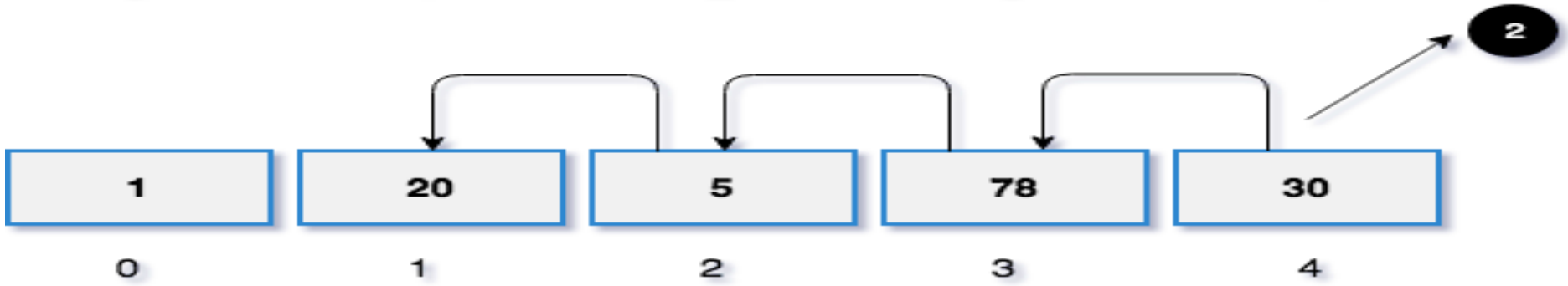
```
Input:  arr[] = {3, 1, 2, 5, 90}, x = 2, size = 5, capacity = 5
Output: arr[] = {3, 1, 5, 90, _}, size = 4, capacity = 5

Input:  arr[] = {3, 1, 2, _, _}, x = 2, size = 3, capacity = 5
Output: arr[] = {3, 1, _, _, _}, size = 4, capacity = 5
```

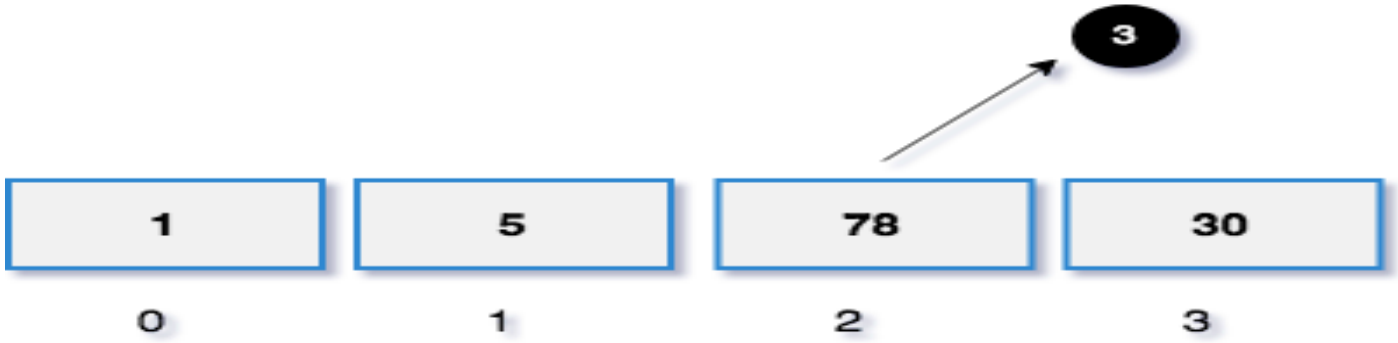
Method 1(Read position, then Remove)

- We first traverse to the position ‘x’ in array at which the array element deletion is required, then move the elements that are on right side of x to one position back.





Deletion of element at position 2



A[1] =A[2]
A[2]=A[3]
A[3]=A[4]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3

Pos =3 means loc is a[2]

Step 1: A[2] =A[3]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	8	8	9	10	11			

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3

Pos =3 means loc is a[2]

Step 1: A[2] =A[3]
Step 2: A[3] =A[4]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	8	9	9	10	11			

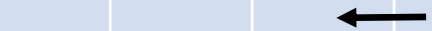
A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3

Pos =3 means loc is a[2]

Step 1: A[2] =A[3]
Step 2: A[3] =A[4]
Step 3: A[4]=A[5]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	8	9	10	10	11			



A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3

Pos =3 means loc is a[2]

Step 1: A[2] =A[3]
Step 2: A[3] =A[4]
Step 3: A[4]=A[5]
Step 4: A[5]=A[6]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	8	9	10	11	11			

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3

Pos =3 means loc is a[2]

Step 1: A[2] =A[3]
Step 2: A[3] =A[4]
Step 3: A[4]=A[5]
Step 4: A[5]=A[6]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	8	9	10	11				

Array size reduced to 5 from 6

Method 1: C program to delete an element from an array – By Specifying Position

- This program deletes or removes an element from an array.
- A user will enter the position at which the array element deletion is required.
- Deleting an element does not affect the size of the array.
- It also checks whether deletion is possible or not, for example, if an array contains five elements and user wants to delete the element at the sixth position, it isn't possible.

```c
#include <stdio.h>
int main()
{
   int array[100], position, c, n;
   printf("Enter number of elements in array\n");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);
   for (c = 0; c < n; c++)
      scanf("%d", &array[c]);
```

Specify how many elements you want.
And Scan those elements

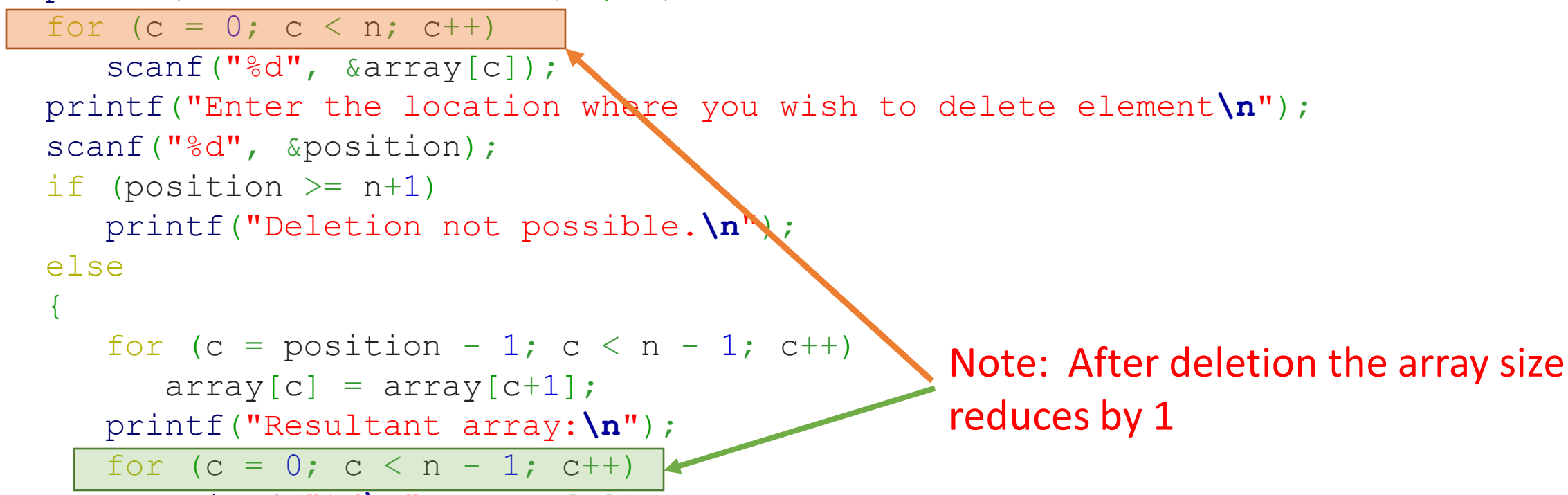
```c
#include <stdio.h>
int main()
{
    int array[100], position, c, n;
    printf("Enter number of elements in array\n");
    scanf("%d", &n);
    printf("Enter %d elements\n", n);
    for (c = 0; c < n; c++)
       scanf("%d", &array[c]);
    printf("Enter the location where you wish to delete element\n");
    scanf("%d", &position);
    if (position >= n+1)
       printf("Deletion not possible.\n");
```

Note: The above if condition checks whether the position of number to be deleted exceeds the array size

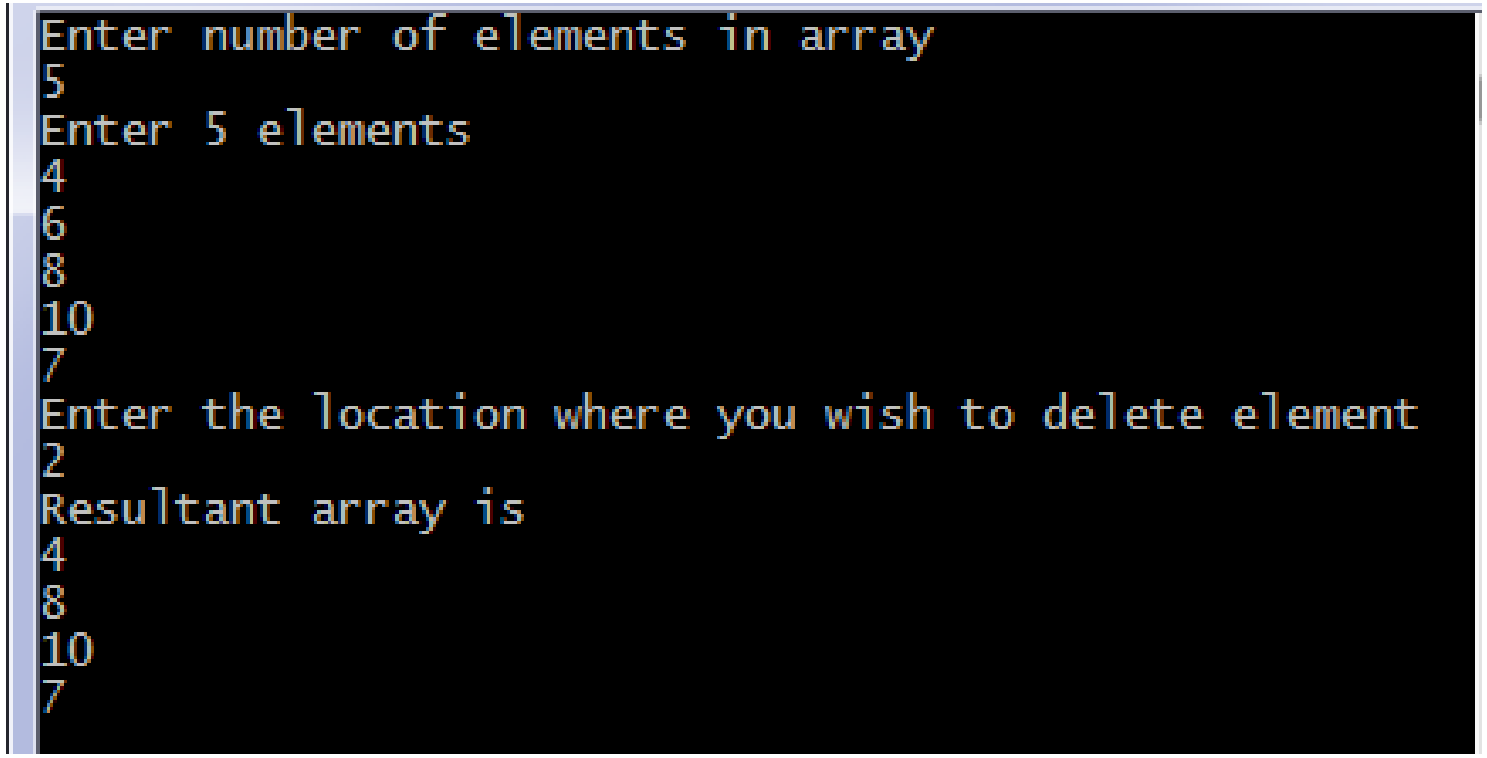
```c
#include <stdio.h>
int main()
{
   int array[100], position, c, n;
   printf("Enter number of elements in array\n");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);
   for (c = 0; c < n; c++)
      scanf("%d", &array[c]);
   printf("Enter the location where you wish to delete element\n");
   scanf("%d", &position);
   if (position >= n+1)
      printf("Deletion not possible.\n");
   else
   {
      for (c = position - 1; c < n - 1; c++)
         array[c] = array[c+1];
```

Note: Shifts array values 1 place to the left so that the deleted cell is overwritten

```c
#include <stdio.h>
int main()
{
   int array[100], position, c, n;
   printf("Enter number of elements in array\n");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);
   for (c = 0; c < n; c++)
      scanf("%d", &array[c]);
   printf("Enter the location where you wish to delete element\n");
   scanf("%d", &position);
   if (position >= n+1)
      printf("Deletion not possible.\n");
   else
   {
      for (c = position - 1; c < n - 1; c++)
         array[c] = array[c+1];
      printf("Resultant array:\n");
      for (c = 0; c < n - 1; c++)
         printf("%d\n", array[c]);
   }
   return 0;
}
```



OUTPUT


Enter number of elements in array
5
Enter 5 elements
4
6
8
10
7
Enter the location where you wish to delete element
2
Resultant array is
4
8
10
7

Method 2(First Search element, then Remove)

- We first read the element ‘x’ which is to be deleted, then search ‘x’ in array, and then move the elements that are on right side of x to one position back.
- Note: if the element is not found, display a relevant message

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	**18**	23	10	31			

Delete value 18

Found is a variable which will be set to 1 if element to be delted is found

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	23 ←	23	10	31			

Step 1: A[3] =A[4]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	**18**	23	10	31			

Delete value 18

Found is a variable which will be set to 1 if element to be delted is found

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	23	10	10	31			

Step 1: A[3] =A[4]
Step 2: A[4] =A[5]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	**18**	23	10	31			

Delete value 18

Found is a variable which will be set to 1 if element to be delted is found

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	23	10	31	31			

Step 1: A[3] =A[4]
Step 2: A[4] =A[5]
Step 3: A[5]=A[6]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	**18**	23	10	31			

Delete value 18

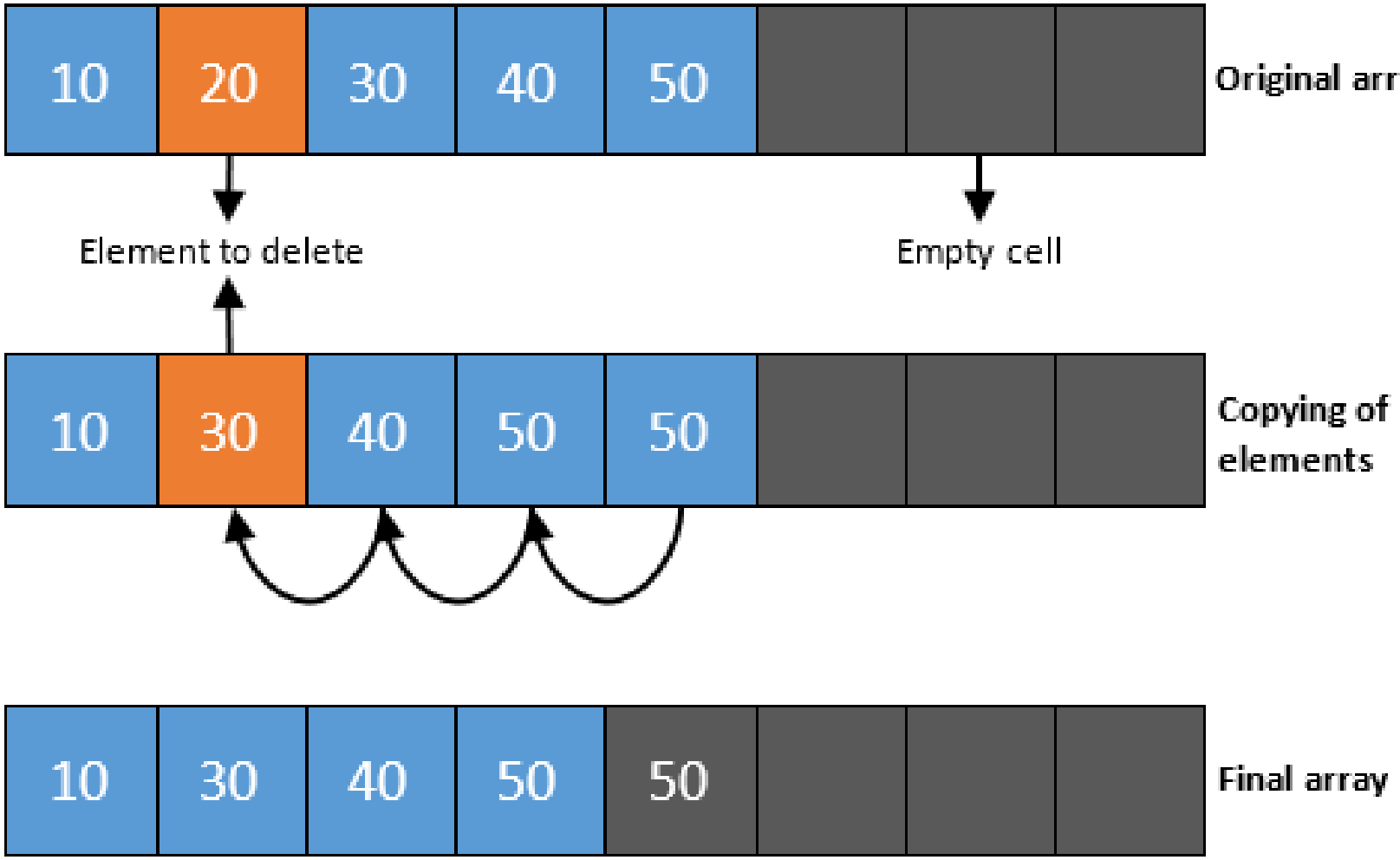
Found is a variable which will be set to 1 if element to be delted is found

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
15	6	9	23	10	31				

Step 1: A[3] =A[4]
Step 2: A[4] =A[5]
Step 3: A[5]=A[6]

Size of Array reduces from 6 to 5

Deletion of Element



```
#include<stdio.h>
int main()
{
int array[10], element, c, n, pos;
int found = 0;
printf("\n\nEnter number of elements in array:");
scanf("%d", &n);
printf("\n\nEnter %d elements\n", n);
for(c = 0; c < n; c++)
       scanf("%d", &array[c]);
printf("\n\nEnter the element to be deleted: ");
scanf("%d", &element);
```

Note: a: Specify the total number of elements
B: Input the elements
C: Specify the element to be deleted

```
#include<stdio.h>
int main()
{int array[10], element, c, n, pos;
int found = 0;
printf("\n\nEnter number of elements in
array:");
scanf("%d", &n);
printf("\n\nEnter %d elements\n", n);
for(c = 0; c < n; c++)
       scanf("%d", &array[c]);
printf("\n\nEnter the element to be deleted:
");
scanf("%d", &element);
// check the element to be deleted is in
array or not
for(c = 0; c < n; c++)
{ if(array[c] == element)
       {
              found = 1;
              pos = c;
              break; // terminate the loop }
}
```

Note: a: Compares every element with the element to be searched

B: Found is a Boolean valued function capable of holding either 0 or 1 where 0 indicates element not in list and 1 indicates it is present

```
#include<stdio.h>
int main()
{int array[10], element, c, n, pos;
int found = 0;
printf("\n\nEnter number of elements in
array:");
scanf("%d", &n);
printf("\n\nEnter %d elements\n", n);
for(c = 0; c < n; c++)
       scanf("%d", &array[c]);
printf("\n\nEnter the element to be deleted:
");
scanf("%d", &element);
// check the element to be deleted is in
array or not
for(c = 0; c < n; c++)
{ if(array[c] == element)
       {
              found = 1;
              pos = c;
              break; // terminate the loop }
}
```

```
if(found == 1)
// the element to be deleted exists in
the array
{
       for(c = pos; c < n-1; c++)
              array[c] = array[c+1];
}
```

Note: Elements will be shifted one place to left only it is found

```
#include<stdio.h>
int main()
{int array[10], element, c, n, pos;
int found = 0;
printf("\n\nEnter number of elements in
array:");
scanf("%d", &n);
printf("\n\nEnter %d elements\n", n);
for(c = 0; c < n; c++)
        scanf("%d", &array[c]);
printf("\n\nEnter the element to be deleted:
");
scanf("%d", &element);
// check the element to be deleted is in
array or not
for(c = 0; c < n; c++)
{ if(array[c] == element)
        {
                found = 1;
                pos = c;
                break; // terminate the loop }
}
```

```
if(found == 1)
// the element to be deleted exists in
the array
{
        for(c = pos; c < n-1; c++)
                array[c] = array[c+1];
}
else
        printf("\n\nElement %d is not
found in the array\n\n", element);
```

Note: If element is not found then no need to shift elements

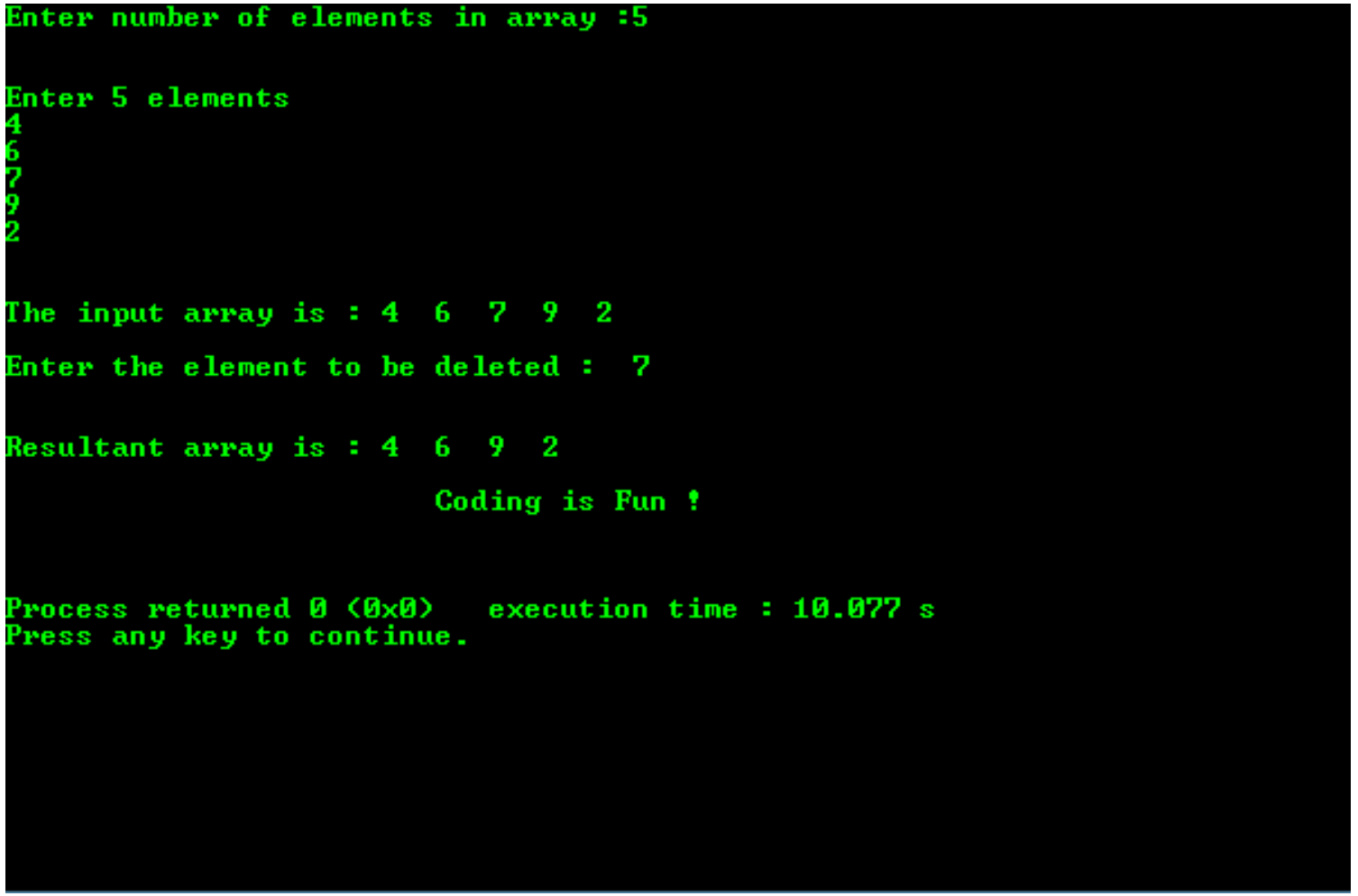
```
#include<stdio.h>
int main()
{int array[10], element, c, n, pos;
int found = 0;
printf("\n\nEnter number of elements in
array:");
scanf("%d", &n);
printf("\n\nEnter %d elements\n", n);
for(c = 0; c < n; c++)
       scanf("%d", &array[c]);
printf("\n\nEnter the element to be deleted:
");
scanf("%d", &element);
// check the element to be deleted is in
array or not
for(c = 0; c < n; c++)
{ if(array[c] == element)
       {
              found = 1;
              pos = c;
              break; // terminate the loop }
}
```

```
if(found == 1)
// the element to be deleted exists in
the array
{
       for(c = pos; c < n-1; c++)
              array[c] = array[c+1];
}
else
       printf("\n\nElement %d is not
found in the array\n\n", element);

printf("\n\nResultant array is: ");
/* the array size gets reduced by 1
after deletion of the element */
for(c = 0; c < n-1; c++)
       printf("%d ",array[c]);
return 0;
}
```

Note: The updated array after deletion

OUTPUT


Enter number of elements in array :5
Enter 5 elements
4
6
7
9
2
The input array is : 4 6 7 9 2
Enter the element to be deleted : 7
Resultant array is : 4 6 9 2
Coding is Fun !
Process returned 0 (0x0) execution time : 10.077 s
Press any key to continue.

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
6	3	2	1	6	6	8	4	5	6

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
6	3	2	1	6	6	8	4	5	6

HINT

1. START SEARCHING FROM RIGHT SIDE OF ARRAY
2. AS SOON AS ELEMENT IS FOUND SAY AT POSITION X THEN FROM X+1 TO N POSITION SHIFT ELEMENTS TO LEFT
3. 3. REDUCE THE SIZE OF ARRAY BY 1 EVERY TIME AN ELEMENT IS FOUND

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
6	3	2	1	6	6	8	4	5	6

Delete Element 6
SIZE OF ARRAY 10

SCAN THE LIST FROM RIGHT TO LEFT
A[9] = 6
SO NOTHING TO BE SHIFTED TO LEFT
REDUCE SIZE OF ARRAY TO 9

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	6	8	4	5	

Delete Element 6
SIZE OF ARRAY 10

SCAN THE LIST FROM RIGHT TO LEFT
A[9] = 6
SO NOTHING TO BE SHIFTED TO LEFT
REDUCE SIZE OF ARRAY TO 9

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	6	8	4	5	

Delete Element 6
SIZE OF ARRAY 9

SCAN THE LIST FROM RIGHT TO LEFT
A[5] = 6

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	8	8	4	5	

Delete Element 6
SIZE OF ARRAY 9

SCAN THE LIST FROM RIGHT TO LEFT
A[5] = 6

Step 1: A[5]=A[6]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	8	4	4	5	

Delete Element 6
SIZE OF ARRAY 9

SCAN THE LIST FROM RIGHT TO LEFT
A[5] = 6

Step 1: A[5]=A[6]
Step 2: A[6] = A[7]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	8	4	5	5	

Delete Element 6
SIZE OF ARRAY 9

SCAN THE LIST FROM RIGHT TO LEFT
A[5] = 6

Step 1: A[5]=A[6]
Step 2: A[6] = A[7]
Step 3: A[7]=A[8]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	
6	3	2	1	6	8	4	5	5	

Delete Element 6
SIZE OF ARRAY 9

SCAN THE LIST FROM RIGHT TO LEFT
A[5] = 6

Step 1: A[5]=A[6]
Step 2: A[6] = A[7]
Step 3: A[7]=A[8]
Reduce Size of Array by 1

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]		
6	3	2	1	6	8	4	5		

Delete Element 6
SIZE OF ARRAY 8

SCAN THE LIST FROM RIGHT TO LEFT

Step 1: A[5]=A[6]
Step 2: A[6] = A[7]
Step 3: A[7]=A[8]
Reduce Size of Array by 1

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]		
6	3	2	1	8	8	4	5		

Delete Element 6
SIZE OF ARRAY 8

SCAN THE LIST FROM RIGHT TO LEFT
A[4]= 6

Step 1: A[4]=A[5]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]		
6	3	2	1	8	4	4	5		

Delete Element 6
SIZE OF ARRAY 8

SCAN THE LIST FROM RIGHT TO LEFT
A[4]= 6

Step 1: A[4]=A[5]
Step 2: A[5]=A[6]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]		
6	3	2	1	8	4	5	5		

Delete Element 6
SIZE OF ARRAY 8

SCAN THE LIST FROM RIGHT TO LEFT
A[4]= 6

Step 1: A[4]=A[5]
Step 2: A[5]=A[6]
Step 3: A[6]=A[7]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
6	3	2	1	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[4]= 6

Step 1: A[4]=A[5]
Step 2: A[5]=A[6]
Step 3: A[6]=A[7]
Reduce Size of Array by 1

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
6	3	2	1	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	3	2	1	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	2	2	1	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	2	1	1	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]
Step 3: A[2]=A[3]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	2	1	8	8	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]
Step 3: A[2]=A[3]
Step 4: A[3]=A[4]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	2	1	8	4	4	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]
Step 3: A[2]=A[3]
Step 4: A[3]=A[4]
Step 5: A[4]=A[5]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]			
3	2	1	8	4	5	5			

Delete Element 6
SIZE OF ARRAY 7

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]
Step 3: A[2]=A[3]
Step 4: A[3]=A[4]
Step 5: A[4]=A[5]
Step 6: A[5]=A[6]

Deletion of an element from an Array

- Delete multiple occurrences of a number from the array

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]				
3	2	1	8	4	5				

Delete Element 6
SIZE OF ARRAY 6

SCAN THE LIST FROM RIGHT TO LEFT
A[0]= 6

Step 1: A[0]=A[1]
Step 2: A[1]=A[2]
Step 3: A[2]=A[3]
Step 4: A[3]=A[4]
Step 5: A[4]=A[5]
Step 6: A[5]=A[6]
Reduce size of array by 1