# Operations on Array

Dr. Asif Uddin Khan

# Basic Operations on Array

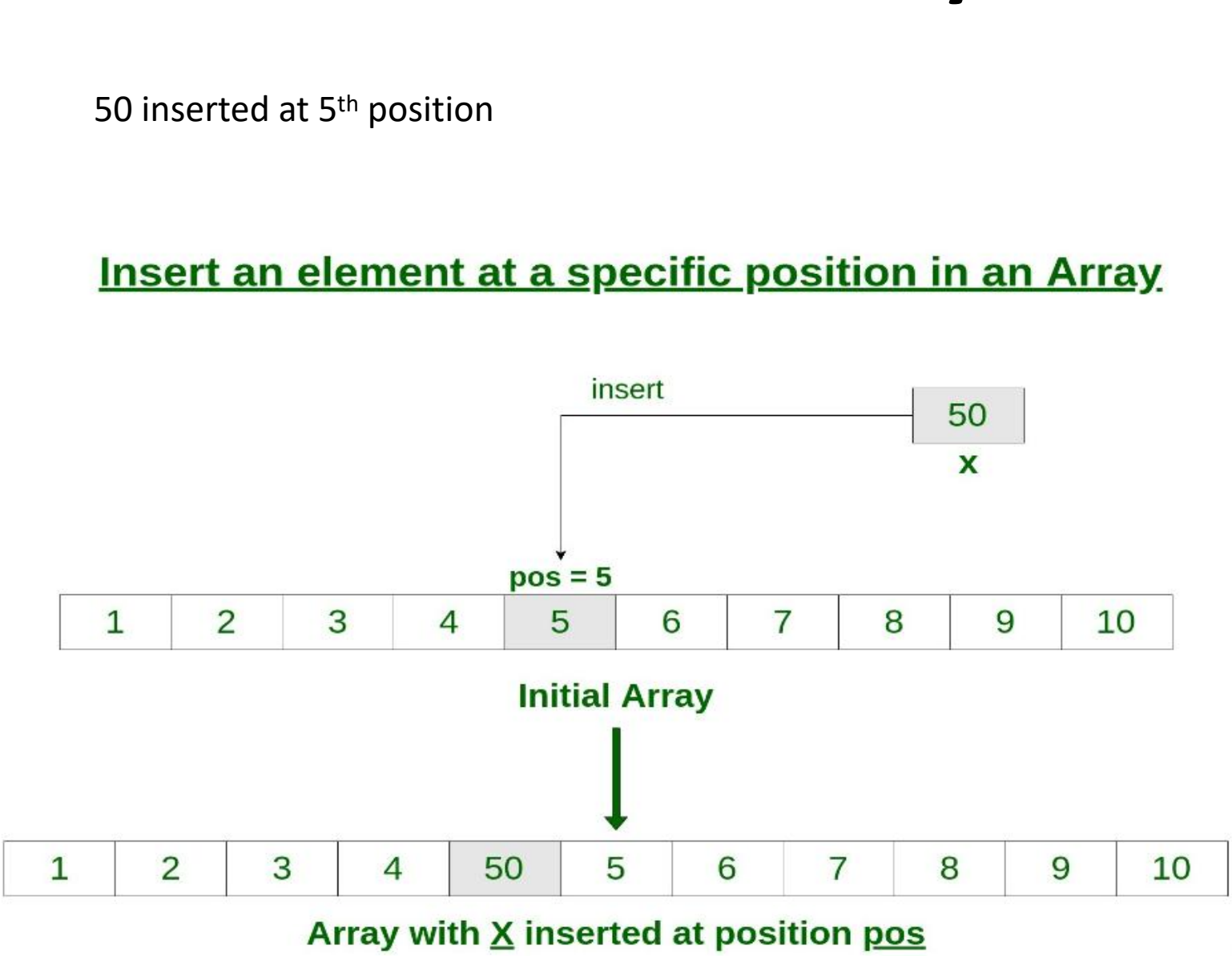
- **Traverse** – Access the array elements so that the data can be checked or used as part of a process.
- **Insertion** – Adds an element at the given index.
- **Deletion** – Deletes an element at the given index.
- **Search** – Searches an element using the given index or by the value.
- **Update** – Updates an element at the given index.
- **Sorting**- Arranging the elements

# Traverse and Print

```
#include<stdio.h>
int main(){
int i=0;
int marks[5]={20,30,40,50,60};
 //traverse and print array elements
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}
return 0;
}
```

# Insertion in Array

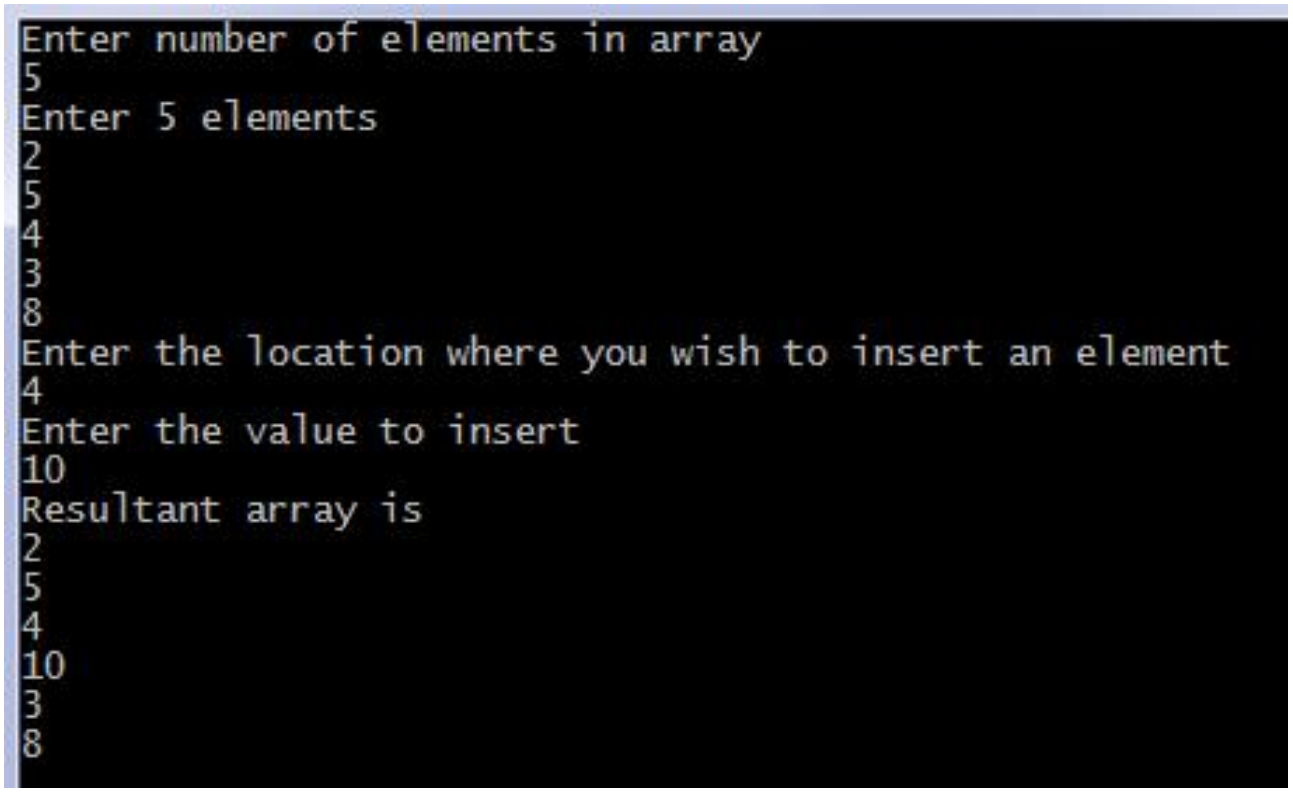
50 inserted at 5th position

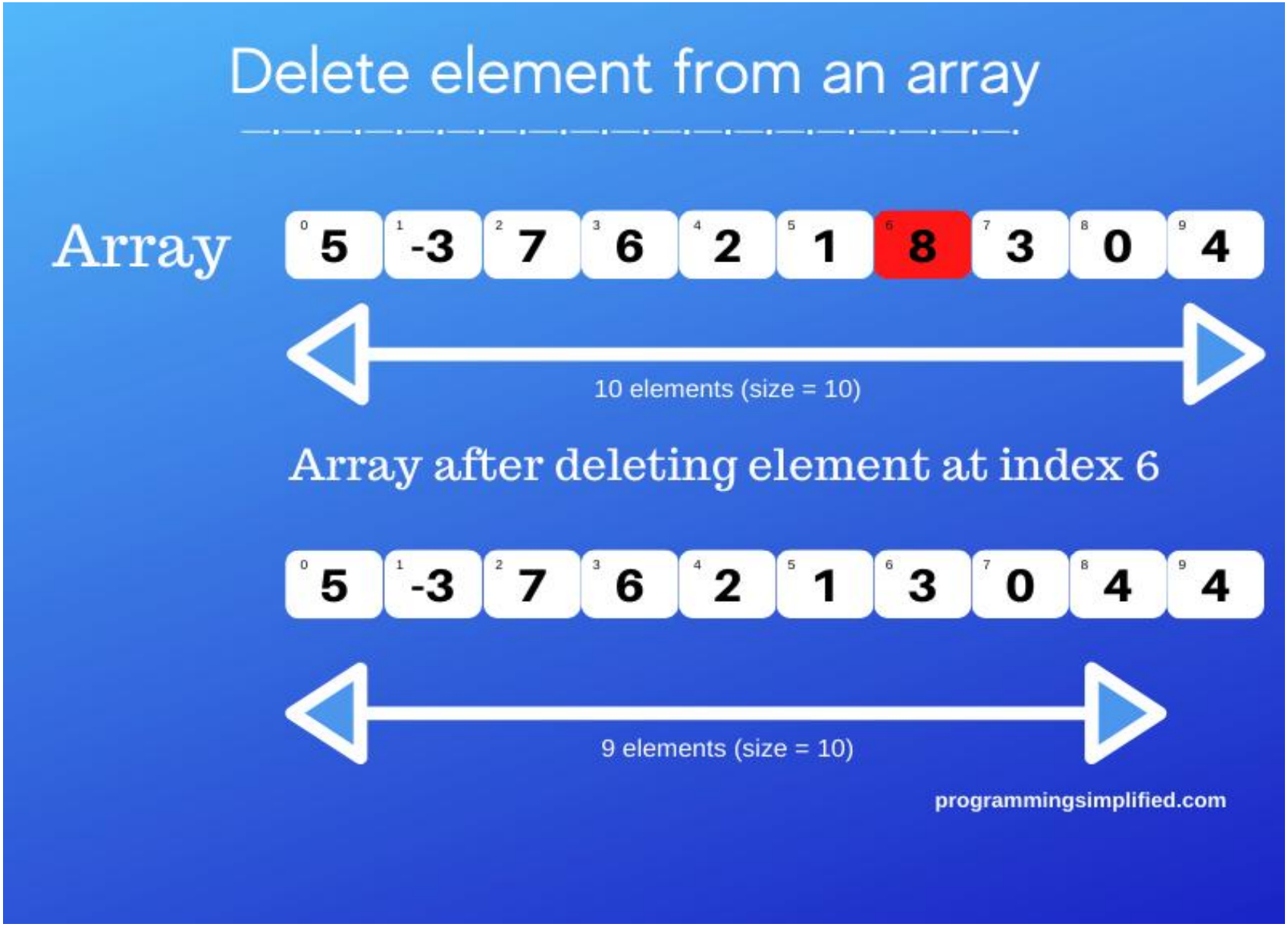


# **Insertion:** Program to insert into array

```
//Program to insert into an array
#include <stdio.h>
int main()
{
   int array[100], position, c, n, value;
   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 insert an element\n");
   scanf("%d", &position);
printf("Enter the value to insert\n");
   scanf("%d", &value);
   for (c = n - 1; c >= position - 1; c--)
      array[c+1] = array[c];
   array[position-1] = value;
   printf("Resultant array is\n");
   for (c = 0; c <= n; c++)
      printf("%d\n", array[c]);
   return 0;
}
```

# Output

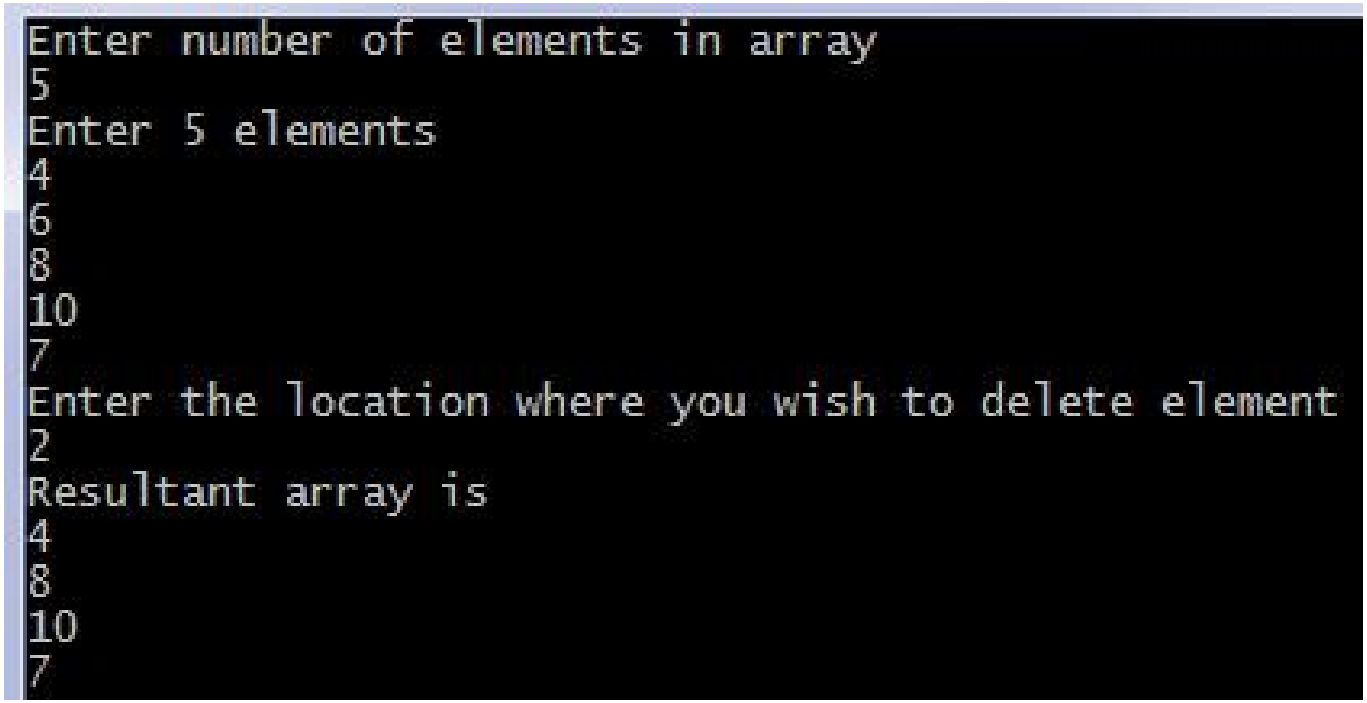

Enter number of elements in array
5
Enter 5 elements
2
5
4
3
8
Enter the location where you wish to insert an element
4
Enter the value to insert
10
Resultant array is
2
5
4
10
3
8


Delete element from an array
Array
0 5
1 -3
2 7
3 6
4 2
5 1
6 8
7 3
8 0
9 4
10 elements (size = 10)
Array after deleting element at index 6
0 5
1 -3
2 7
3 6
4 2
5 1
6 3
7 0
8 4
9 4
9 elements (size = 10)
programmingsimplified.com

# Program remove element from array C

```
//delete from array
#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

# Sorting Descending order

```c
// sorting in array using bubble sorting
#include<stdio.h>
void main ()
{
    int i, j,temp;
    int a[10] = { 10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
    for(i = 0; i<10; i++)
    {
        for(j = i+1; j<10; j++)
        {
            if(a[j] > a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("Printing Sorted Element List ...\n");
    for(i = 0; i<10; i++)
    {
        printf("%d\n",a[i]);
    }
}
```

# Sorting in ascending order

```c
// sorting in array using bubble sorting
#include<stdio.h>
void main ()
{
    int i, j,temp;
    int a[10] = { 10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
    for(i = 0; i<10; i++)
    {
        for(j = i+1; j<10; j++)
        {
            if(a[j] < a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("Printing Sorted Element List ...\n");
    for(i = 0; i<10; i++)
    {
        printf("%d\n",a[i]);
    }
}
```