

# Insertion Sort

Insertion sort is a simple sorting algorithm that works the way we sort playing cards in our hands.

## Algorithm

// Sort an arr[] of size n

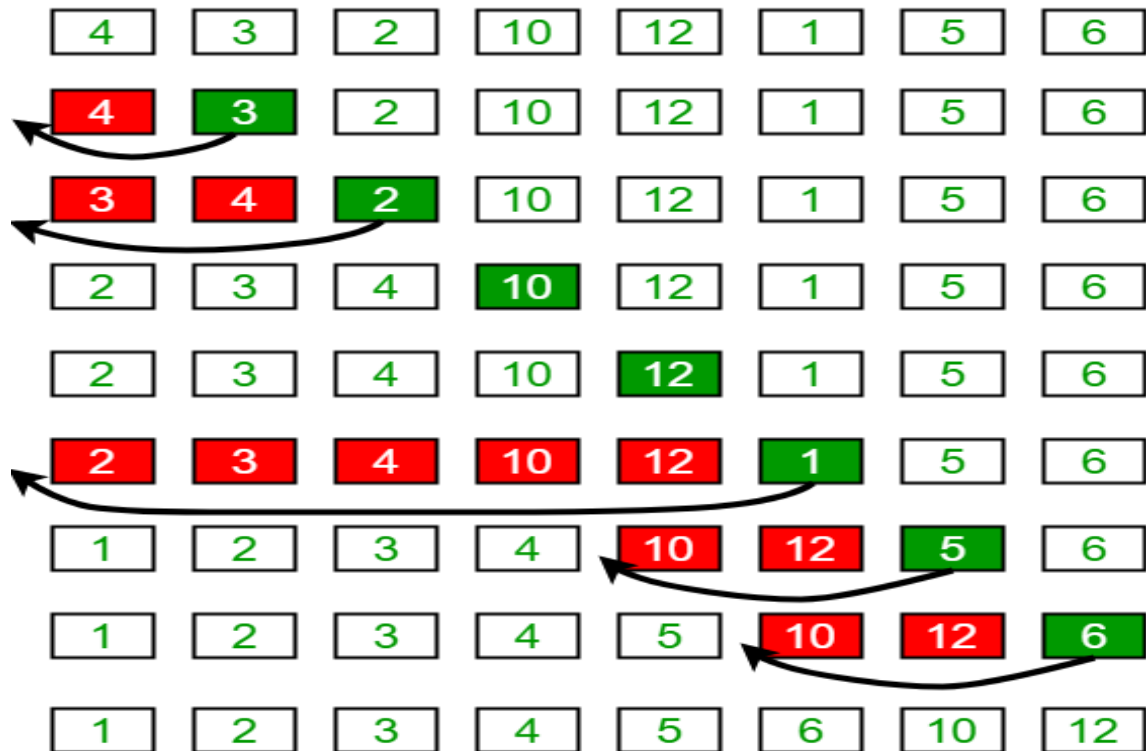
insertionSort(arr, n)

Loop from i = 1 to n-1.

.....a) Pick element arr[i] and insert it into sorted sequence arr[0...i-1]

Example:

## Insertion Sort Execution Example



```
#include <math.h>
```

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

```
/* Function to sort an array using insertion sort*/
```

```
void insertionSort(int *arr, int n)
```

```
{
```

```
    int i, key, j;
```

```
    /* i is used to mark the key element */
```

```
    /* Partially sorted array arr[0..i-1]*/
```

```
    for (i = 1; i < n; i++)
```

```
    {
```

```
        key = arr[i];
```

```
        j = i - 1;
```

```
        /* Move elements of arr[0..i-1], that are  
        greater than key, to one position ahead  
        of their current position */
```

```
        while (j >= 0 && arr[j] > key) {
```

```
            arr[j + 1] = arr[j];
```

```
            j = j - 1;
```

```
        }
```

```
        arr[j + 1] = key;
```

```
    }
```

```

}

// A utility function to print an array of size n
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d  ", arr[i]);
    printf("\n");
}

/* Driver program to test insertion sort */
int main()
{
    int arr[] = { 12, 11, 13, 5, 6 };
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("\n Content of the array before sorting:\n");
    printArray(arr, n);

    insertionSort(arr, n);
    printf("\n Content of the array after sorting:\n");
    printArray(arr, n);

    return 0;
}

```