

PRACTICAL:-02

AIM:- WAP and algorithm of insertion sorts

//ALGORITHM

The simple steps of achieving the insertion sort are listed as follows –

Step 1 - If the element is the first element, assume that it is already sorted. Return 1.

Step2 - Pick the next element, and store it separately in a key.

Step3 - Now, compare the key with all elements in the sorted array. Step

4 - If the element in the sorted array is smaller than the current element, then move to

the next element. Else, shift greater elements in the array towards the right.

Step 5 - Insert the value.

Step 6 - Repeat until the array is sorted.

// COMPEXITY

Best Case	$O(n)$
Average Case	$O(n^2)$
Worst Case	$O(n^2)$

//PROGRAM

#include <math.h>

#include <stdio.h>

void insertionSort(int arr[], int n)

{

int i, key, j;

```

    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;

        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

```

```

void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

```

```

int main()
{
    int arr[] = { 11, 20, 32, 8, 1 };
    int n = sizeof(arr) / sizeof(arr[0]);

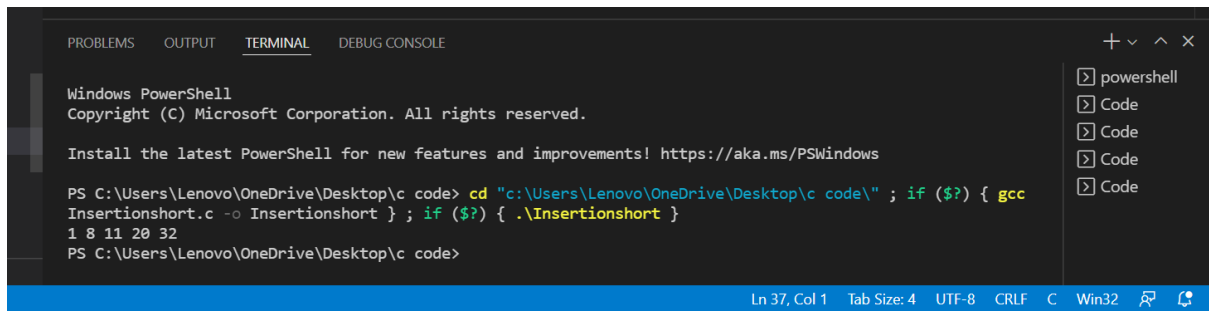
    insertionSort(arr, n);
}

```

```
    printArray(arr, n);

    return 0;
}
```

//OUTPUT



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Lenovo\OneDrive\Desktop\c code> cd "c:\Users\Lenovo\OneDrive\Desktop\c code\" ; if ($?) { gcc
Insertionshort.c -o Insertionshort } ; if ($?) { .\Insertionshort }
1 8 11 20 32
PS C:\Users\Lenovo\OneDrive\Desktop\c code>
```