**Program no:-3**

**Objective**:- Implementation of sorting techniques( Insertion sort).

**Theory:**- Insertion sort is a sorting algorithm where the array is sorted by taking one element at a time. The principle behind insertion is to take one element, iterate through the sorted array & find its correct position in the sorted array. Insertion sort works in a similar manner as we arrange a deck of cards.

# **Algorithm:-**

**Step 1** − If the element is the first one, it is already sorted.

**Step 2** – Move to next element

**Step 3** − Compare the current element with all elements in the sorted array

**Step 4** – If the element in the sorted array is smaller than the current element, iterate to the next element. Otherwise, shift all the greater element in the array by one position towards the right

**Step 5** − Insert the value at the correct position

**Step 6** − Repeat until the complete list is sorted

**Code:**

#include <stdio.h>

void swap(int \*xp, int \*yp) {

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void selectionSort(int arr[], int n) {

int i, j, min\_idx;

for (i = 0; i < n - 1; i++) {

min\_idx = i;

for (j = i + 1; j < n; j++) {

if (arr[j] < arr[min\_idx]) {

min\_idx = j;

}

}

swap(&arr[min\_idx], &arr[i]);

}

}

int main() {

printf("PIYUSH MITTAL, 2201920100208");

printf("\n");

int n, i;

printf("Enter the number of elements: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d integers:\n", n);

for (i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

printf("\n");

selectionSort(arr, n);

printf("Sorted array: ");

for (i = 0; i < n; i++) {

printf("%d ", arr[i]);

printf("\n");

}

return 0;

}

**OUTPUT:**

