**Practical-5**

**AIM:** To implement three types of sorting in C programming.

1. Selection Sort
2. Insertion Sort
3. Bubble sort

**SOFTWARE REQUIRED:** Vs Code

**PSEUDO CODE:**

**Selection Sort:**

procedure selection sort

list : array of items

n : size of list

for i = 1 to n - 1

/\* set current element as minimum\*/

min = i

/\* check the element to be minimum \*/

for j = i+1 to n

if list[j] < list[min] then

min = j;

end if

end for

/\* swap the minimum element with the current element\*/

if indexMin != i then

swap list[min] and list[i]

end if

end for

end procedure

**Insertion Sort:**

procedure insertionSort( A : array of items )

int holePosition

int valueToInsert

for i = 1 to length(A) inclusive do:

/\* select value to be inserted \*/

valueToInsert = A[i]

holePosition = i

/\*locate hole position for the element to be inserted \*/

while holePosition > 0 and A[holePosition-1] > valueToInsert do:

A[holePosition] = A[holePosition-1]

holePosition = holePosition -1

end while

/\* insert the number at hole position \*/

A[holePosition] = valueToInsert

end for

end procedure

**Bubble Sort:**

bubbleSort(array)

for i <- 1 to indexOfLastUnsortedElement-1

if leftElement > rightElement

swap leftElement and rightElement

end bubbleSort

**CODE:**

**a) Selection Sort**

#include <stdio.h>

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

    for (int i = 0; i < n - 1; i++) {

        int minimum = i;

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

            if (arr[j] < arr[minimum]) {

                minimum = j;

            }

        }

        if (minimum != i) {

            int temp = arr[i];

            arr[i] = arr[minimum];

            arr[minimum] = temp;

        }

    }

}

int main() {

    int n;

    printf("Name: Ananta Walli");

    printf("\nEnrollment Number: A23065221322");

    printf("\nPlease enter the number of elements that are needed for sorting: ");

    scanf("%d", &n);

    int arr[n];

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

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    function(arr, n);

   printf("\nThe elements after selection sort will be:\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

**b)Insertion Sort**

#include <stdio.h>

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

    for (int i = 1; i < n; i++) {

        int key = arr[i];

        int j = i - 1;

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

            arr[j + 1] = arr[j];

            j--;

        }

        arr[j + 1] = key;

    }

}

int main() {

    int n;

    printf("Name: Ananta Walli");

    printf("\nEnrollment No: A2305221322");

    printf("\nPlease enter the number of elements: ");

    scanf("%d", &n);

    int arr[n];

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

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    function(arr, n);

    printf("\nThe insertion sort will give result:\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

**c)Bubble Sort**

#include <stdio.h>

void swap(int \*x, int \*y) {

    int temp = \*x;

    \*x = \*y;

    \*y = temp;

}

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

    for (int i = 0; i < n - 1; i++) {

        int swapped = 0;

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

            if (arr[j] > arr[j + 1]) {

                swap(&arr[j], &arr[j + 1]);

                swapped = 1;

            }

        }

        if (swapped == 0)

            break;

    }

}

int main() {

    int n;

    printf("Name: Ananta Walli");

    printf("\nEnrollment Number: A23065221322");

    printf("\nPlease enter the number of elements: ");

    scanf("%d", &n);

    int arr[n];

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

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    function(arr, n);

    printf("\nThe elements after bubble sort will be:\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

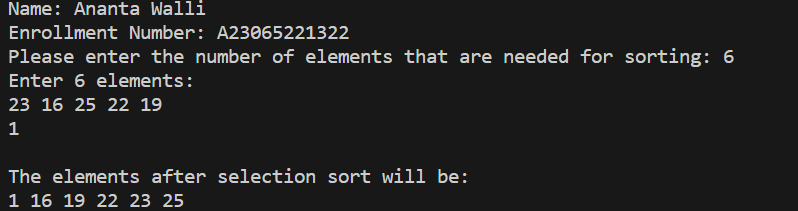
    printf("\n");

    return 0;

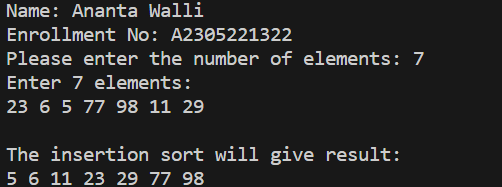
}

**OUTPUT:**

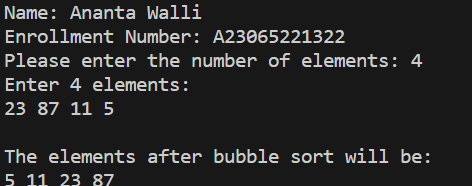
**Selection Sort:**



**Insertion Sort:**

****

**Bubble Sort:**



**TIME COMPLEXITY:**

1. **Selection Sort:**

Average Case: O(n^2)

Best Case: O(n^2)

Worst Case: O(n^2)

1. **Insertion Sort:**

Average Case: O(n^2)

Best Case: O(n)

Worst Case: O(n^2)

1. **Bubble Sort:**

Average Case: O(n^2)

Best Case: O(n)

Worst Case: O(n^2)

**RESULT:** The above code implements the Selection Sort, Insertion sort and Bubble Sort in C programming.