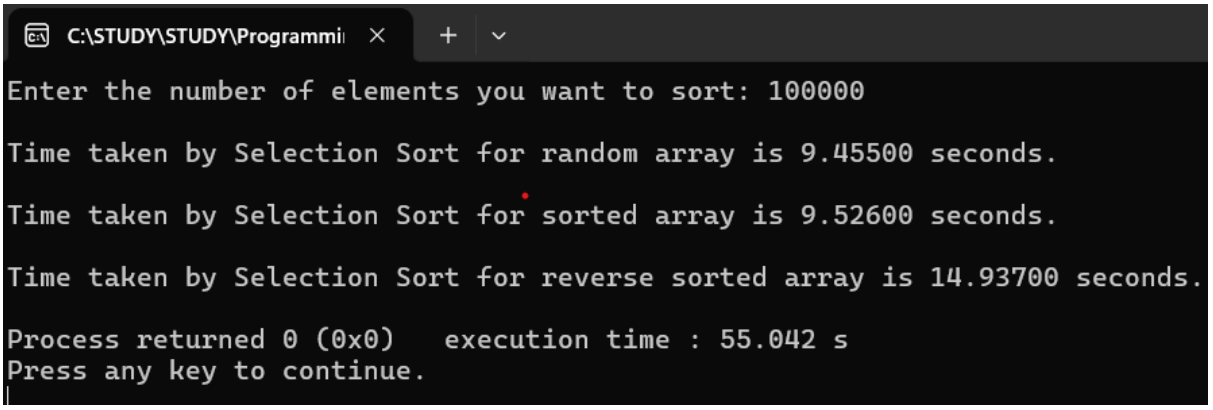
# PRACTICAL 2

**AIM: Write a program to sort given elements of an array in ascending order using Selection sort. Analyze the time complexity for best, average and worst case.**

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

void reverseArray(int* arr, int size) {
   int start = 0;
   int end = size - 1;
   while (start < end) {
      int temp = arr[start];
      arr[start] = arr[end];
      arr[end] = temp;
      start++;
      end--;
   }
}

void selectionSort(int* arr, int size) {
   int smallest, temp;
   for (int j = 0; j < (size - 1); j++) {
      smallest = j;
      for (int i = (j + 1); i < size; i++) {
         if (arr[i] < arr[smallest]) {
            smallest = i;
         }
      }
      temp = arr[j];
      arr[j] = arr[smallest];
      arr[smallest] = temp;
   }
}

int main() {
   int SIZE;
   printf("Enter the number of elements you want to sort: ");
```

```
    scanf("%d", &SIZE);

    int* arr = (int*)malloc(SIZE * sizeof(int));
    for (int i = 0; i < SIZE; i++) {
        arr[i] = rand() % SIZE;
    }

    clock_t start, end;
    double cpu_time;

    // Sort the array and measure the time
    start = clock();
    selectionSort(arr, SIZE);
    end = clock();
    cpu_time = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("\nTime taken by Selection Sort for random array is %.5f seconds.\n", cpu_time);

    clock_t start1, end1;
    double cpu_time1;
    start1 = clock();
    selectionSort(arr , SIZE);
    end1 = clock();
    cpu_time1 = ((double)(end1 - start1)) / CLOCKS_PER_SEC;
    printf("\nTime taken by Selection Sort for sorted array is %.5f seconds.\n", cpu_time1);

    // Reverse the sorted array
    reverseArray(arr, SIZE);

    // Measure the time to sort the reverse sorted array
    clock_t start2, end2;
    double cpu_time2;
    start2 = clock();
    selectionSort(arr, SIZE);
    end2 = clock();
    cpu_time2 = ((double)(end2 - start2)) / CLOCKS_PER_SEC;
    printf("\nTime taken by Selection Sort for reverse sorted array is %.5f seconds.\n",
cpu_time2);

    free(arr);
    return 0;
}
```

# OUTPUT:




Output for different size of input

| | 30000 | 50000 | 80000 | 100000 | 150000 |
|---|---|---|---|---|---|
| Random Array | 0.84100 | 2.32500 | 5.94600 | 9.455000 | 20.91400 |
| Sorted Array | 0.84600 | 2.31700 | 6.03900 | 9.526000 | 20.93100 |
| Reverse Sorted Array | 1.16500 | 3.83900 | 9.33200 | 14.93700 | 28.76800 |

Where elapsed time is in seconds.