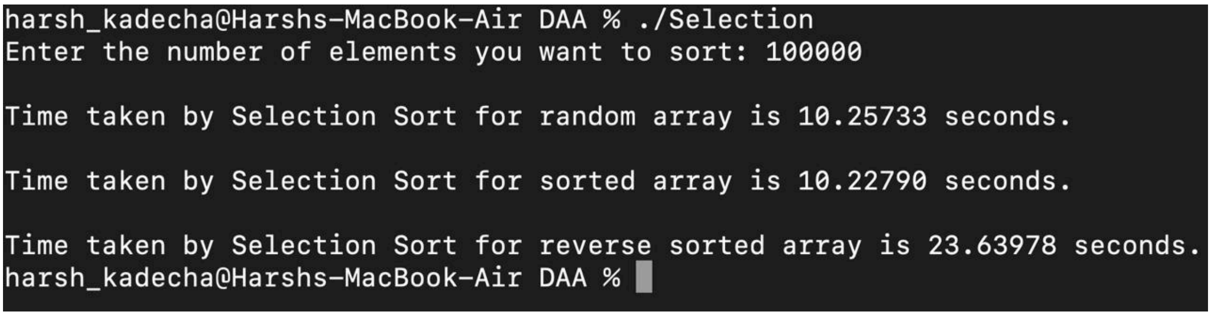
PRACTICAL 2

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:



```
harsh_kadecha@Harshs-MacBook-Air DAA % ./Selection
Enter the number of elements you want to sort: 100000

Time taken by Selection Sort for random array is 10.25733 seconds.

Time taken by Selection Sort for sorted array is 10.22790 seconds.

Time taken by Selection Sort for reverse sorted array is 23.63978 seconds.
harsh_kadecha@Harshs-MacBook-Air DAA %
```

Output for different size of input

	30000	50000	80000	100000	150000
Random Array	0.94080	2.57900	6.58607	10.25733	23.05572
Sorted Array	0.92075	2.55858	6.54792	10.22790	23.01249
Reverse Sorted Array	2.04238	5.75887	15.02191	23.63978	53.03392

Where elapsed time is in seconds.