DSA – Practical: Sorting

Refer to the lecture notes for sorting algorithms and work on the following tasks:

1. Write the code of the two functions: bubbleSort() and selectSort():

```
// Sort array of integers in ascending order using bubble sort algorithm
void bubbleSort(int data[], int size){
    ...
}

// Sort array of integers in ascending order using selection sort algorithm
void selectionSort(int data[], int size){
    ...
}
```

2. Test the algorithms are working correctly by sorting and printing sample arrays (before and after sorting).

```
#define MAX_SIZE 10

void main () {

   int arr1[] = {4, 23, 8, 0, -5, 12, 1, 33, 6, 20};
   int arr2[] = {4, 23, 8, 0, -5, 12, 1, 33, 6, 20};

   printf("Arr1 before sorting:\n");
   printArray(arr1, MAX_SIZE);

   bubbleSort(arr1, MAX_SIZE);

   printf("Arr1 after sorting with bubble sort:\n");
   printArray(arr1, MAX_SIZE);

   ...

   system("pause");
}
```

3. Create much bigger arrays (for example, `MAX_SIZE 30000`) and fill them with random integer numbers. Include the library “time.h” and refer to the functions srand() and rand() and use them to generate random numbers filling both arrays.

Hint: `arr1[i] = arr2[i] = rand() % 1000;  //fill values between 0 and 999`

4. Measure the time needed to sort the same random arrays with each sorting algorithm and compare the results. Time cost can be measured in seconds using the time() function:

```
time_t timeBefore, timeAfter, timeNeededInSeconds;
timeBefore = time(NULL);

bubbleSort(arr1, MAX_SIZE);

timeAfter = time(NULL);
timeNeededInSeconds = timeAfter - timeBefore;
printf("Time to sort using bubble sort was: %d seconds.\n\n", timeNeededInSeconds);
```

- The End -