#include<iostream>

using namespace std;

void findMinIndex(int arr[], int size, int curr, int &index)

{

if (curr <= size - 1) //base case will be met when curr will be greater than or equal to size

{

//an if statement to compare minimum val with every remaining index

if (arr[curr] < arr[index])

{

index = curr; //update index if value is found

}

findMinIndex(arr, size, ++curr, index); //recursive call

}

else

return;

}

void selectionSort(int arr[], int size) //two parameters to keep it generic

{

static int i = 0; //variable to traverse the array

if (i == size - 1 + i) //base case: return when the whole array has been traversed

return;

else

{

int index = i + 1; //Supposing whatever is at arr[i+1] is minimum

findMinIndex(arr, size + i, i + 1, index); //index will be updated after the function call

//size+i because size is being decremented after every function call

swap(arr[i], arr[index]), ++i; //tradition swap op. and increment in i

selectionSort(arr, --size); //recursive call

}

}

void print(int arr[], int size)

{

for (int i = 0; i < size; ++i) //printing array elements (space separated)

cout << arr[i] << " ";

cout << endl;

}

int main()

{

/\*Declaring an array and filling it with random values\*/

const int size = 10;

int arr[size];

for (int i = 0; i < size; ++i)

arr[i] = rand() % 10 + 1;

/\*Printing the array with random values\*/

print(arr, size);

/\*Calling the function\*/

selectionSort(arr, size);

/\*Printing the sortred array\*/

print(arr, size);

system("pause");

}