***5)*** Implementing Selection Sort using Function Templates in C++

***CODE :-***

#include <iostream>

using namespace std;

template <typename T>

void selectionSort(T arr[], int size)

{

int minIndex;

T temp;

for (int i = 0; i < size - 1; i++)

{

minIndex = i;

for (int j = i + 1; j < size; j++)

{

if (arr[j] < arr[minIndex])

{

minIndex = j;

}

}

if (minIndex != i)

{

temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

}

template <typename T>

void printArray(T arr[], int size)

{

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

{

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

}

cout << endl;

}

int main()

{

int intArr[] = {29, 10, 14, 37, 13};

int intSize = sizeof(intArr) / sizeof(intArr[0]);

cout << "Original integer array: ";

printArray(intArr, intSize);

selectionSort(intArr, intSize);

cout << "Sorted integer array: ";

printArray(intArr, intSize);

float floatArr[] = {3.4, 2.1, 5.6, 1.9, 4.7};

int floatSize = sizeof(floatArr) / sizeof(floatArr[0]);

cout << "Original float array: ";

printArray(floatArr, floatSize);

selectionSort(floatArr, floatSize);

cout << "Sorted float array: ";

printArray(floatArr, floatSize);

return 0;

}

***OUTPUT :-***

