Câu hỏi 3

Chính xác

Chấm điểm của 2,00

Implement static methods **sortSegment** and **ShellSort** in class **Sorting** to sort an array in ascending order.

```
#ifndef SORTING_H
#define SORTING_H

#include <sstream>
#include <iostream>
#include <type_traits>
using namespace std;

template <class T>
class Sorting {
private:
    static void printArray(T* start, T* end)
    {
        int size = end - start;
        for (int i = 0; i < size; i++)
            cout << start[i] << " ";
        cout << endl;
    }

public:
    // TODO: Write your code here
    static void sortSegment(T* start, T* end, int segment_idx, int cur_segment_total);
    static void ShellSort(T* start, T* end, int* num_segment_list, int num_phases);
};
```

```
#endif /* SORTING_H */
```

For example:

Test	Result

int num_segment_list[] = {1, 3, 5};
int num_phases = 3;
int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };
 	
Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);
	5 segments: 5 4 3 2 1 10 9 8 7 6
	3 segments: 2 1 3 5 4 7 6 8 10 9
	1 segments: 1 2 3 4 5 6 7 8 9 10

Answer: (penalty regime: 0 %)

Reset answer

```
// TODO: Write your code here

static void sortSegment(T* start, T* end, int segment_idx, int
    // TODO
    int size = end - start;
    int key, j;
    for (int i = 1; i * cur_segment_total + segment_idx < size
        key = start[i * cur_segment_total + segment_idx];
        j = i - 1;

        while (j >= 0 && start[j * cur_segment_total + segment
            start[(j + 1) * cur_segment_total + segment_idx] =
            j = j - 1;
        }

        start[(j + 1) * cur_segment_total + segment_idx] = key
    }
}
```

```
static void ShellSort(T* start, T* end, int* num_segment_list,
    // TODO
    // Note: You must print out the array after sorting segmen
    for (int i = num_phases - 1; i >= 0; i--){
        for (int segment = 0; segment < num_segment_list[i]; +
            sortSegment(start, end, segment, num_segment_list[
        }
        cout << num_segment_list[i] << " segments: ";
        printArray(start, end);
    }
}
```

Precheck   Kiểm tra

	Test	Expected	Got	

✓	int num_segment_list[] = {1, 3, 5};
	int num_phases = 3;
	int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };
	
	Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);
		5 segments: 5 4 3 2 1 10 9 8 7 6
		3 segments: 2 1 3 5 4 7 6 8 10 9
		1 segments: 1 2 3 4 5 6 7 8 9 10
			5 segments: 5 4 3 2 1 10 9 8 7 6
			3 segments: 2 1 3 5 4 7 6 8 10 9
			1 segments: 1 2 3 4 5 6 7 8 9 10
				✓

✓	int num_segment_list[] = { 1, 2, 6 };
	int num_phases = 3;
	int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };
	
	Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);
		6 segments: 4 3 2 1 6 5 10 9 8 7
		2 segments: 2 1 4 3 6 5 8 7 10 9
		1 segments: 1 2 3 4 5 6 7 8 9 10
			6 segments: 4 3 2 1 6 5 10 9 8 7
			2 segments: 2 1 4 3 6 5 8 7 10 9
			1 segments: 1 2 3 4 5 6 7 8 9 10
				✓

✓	int num_segment_list[] = { 1, 2, 5 };
	int num_phases = 3;
	int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };
	
	Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);
		5 segments: 5 4 3 2 1 10 9 8 7 6
		2 segments: 1 2 3 4 5 6 7 8 9 10
		1 segments: 1 2 3 4 5 6 7 8 9 10
			5 segments: 5 4 3 2 1 10 9 8 7 6
			2 segments: 1 2 3 4 5 6 7 8 9 10
			1 segments: 1 2 3 4 5 6 7 8 9 10
				✓

✓	int num_segment_list[] = { 1, 2, 3 };
	int num_phases = 3;
	int array[] = { 10, 9, 8 , 7 , 6, 5, 4, 3, 2, 1 };
	
	Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);
		3 segments: 1 3 2 4 6 5 7 9 8 10
		2 segments: 1 3 2 4 6 5 7 9 8 10
		1 segments: 1 2 3 4 5 6 7 8 9 10
			3 segments: 1 3 2 4 6 5 7 9 8 10
			2 segments: 1 3 2 4 6 5 7 9 8 10
			1 segments: 1 2 3 4 5 6 7 8 9 10
				✓

✓	int num_segment_list[] = { 1, 5, 8, 10 };
	int num_phases = 4;
	int array[] = { 3, 5, 7, 10 ,12, 14, 15, 13, 1, 2, 9, 6, 4, 8, 11 };
	
	Sorting<int>::ShellSort(&array[0], &array[15], &num_segment_list[0], num_phases);
		10 segments: 3 5 4 8 11 14 15 13 1 2 9 6 7 10 12
		8 segments: 1 2 4 6 7 10 12 13 3 5 9 8 11 14 15
		5 segments: 1 2 4 3 5 9 8 11 6 7 10 12 13 14 15
		1 segments: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
			10 segments: 3 5 4 8 11 14 15 13 1 2 9 6 7 10 12
			8 segments: 1 2 4 6 7 10 12 13 3 5 9 8 11 14 15
			5 segments: 1 2 4 3 5 9 8 11 6 7 10 12 13 14 15
			1 segments: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
				✓

Passed all tests! ✓

BÁCH KHOA E-LEARNING

WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn