Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Ba, 25 tháng 2 2025, 1:57 PM
Kết thúc lúc	Thứ Hai, 10 tháng 3 2025, 11:58 PM
Thời gian thực hiện	13 Các ngày 10 giờ
Điểm	1,10/3,00
Điểm	**3,67** trên 10,00 (**36,67**%)

Câu hỏi 1

Đúng một phần

Đạt điểm 0,10 trên 1,00

Implement method bubbleSort() in class SLinkedList to sort this list in ascending order. After each bubble, we will print out a list to check (using printList).

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

template <class T>
class SLinkedList {
public:
    class Node; // Forward declaration
protected:
    Node* head;
    Node* tail;
    int count;
public:
    SLinkedList()
    {
      this->head = nullptr;
      this->tail = nullptr;
      this->count = 0;
    }
    ~SLinkedList(){};
    void add(T e)
    {
        Node *pNew = new Node(e);

        if (this->count == 0)
        {
            this->head = this->tail = pNew;
        }
        else
        {
            this->tail->next = pNew;
            this->tail = pNew;
        }

        this->count++;
    }
    int size()
    {
        return this->count;
    }
    void printList()
    {
        stringstream ss;
        ss << "[";
        Node *ptr = head;
        while (ptr != tail)
        {
            ss << ptr->data << ",";
            ptr = ptr->next;
        }

        if (count > 0)
            ss << ptr->data << "]";
        else
            ss << "]";
        cout << ss.str() << endl;
    }
public:
    class Node {
    private:
        T data;
        Node* next;
        friend class SLinkedList<T>;
    public:
        Node() {
            next = 0;
        }
        Node(T data) {
            this->data = data;
            this->next = nullptr;
        }
```

```
    };

    void bubbleSort();
};
```

For example:

Test	Result

int arr[] = {9, 2, 8, 4, 1};
SLinkedList<int> list;
for(int i = 0; i <int(sizeof(arr))/4;i++)
    list.add(arr[i]);
list.bubbleSort();
	[2,8,4,1,9]
	[2,4,1,8,9]
	[2,1,4,8,9]
	[1,2,4,8,9]

Answer: (penalty regime: 0 %)

Reset answer

```
template <class T>
void SLinkedList<T>::bubbleSort() {
    if (!head || !head->next) return; // Nếu danh sách rỗng hoặc có một phần tử, không cần sắp xếp

    bool swapped;
    Node* ptr;
    Node* lastSorted = nullptr; // Giới hạn phạm vi kiểm tra sau mỗi lần lặp

    do {
        swapped = false;
        ptr = head;

        while (ptr->next != lastSorted) {
            if (ptr->data > ptr->next->data) {
                swap(ptr->data, ptr->next->data);
                swapped = true;
            }
            ptr = ptr->next;
        }
        lastSorted = ptr; // Giảm phạm vi kiểm tra dần dần

        printList(); // In danh sách sau mỗi lần đổi chỗ
    } while (swapped);
}

```

	Test	Expected	Got	

✗	`int arr[] = {9, 2, 8, 4, 1};`
	`SLinkedList<int> list;`
	`for(int i = 0; i <int(sizeof(arr))/4;i++)`
	`  list.add(arr[i]);`
	`list.bubbleSort();`
		[2,8,4,1,9]
		[2,4,1,8,9]
		[2,1,4,8,9]
		[1,2,4,8,9]
			[2,8,4,1,9]
			[2,4,1,8,9]
			[2,1,4,8,9]
			[1,2,4,8,9]
			[1,2,4,8,9]
				✗

Some hidden test cases failed, too.

Show differences

Đúng một phần

Marks for this submission: 0,10/1,00.

Câu hỏi **2**

Đúng

Đạt điểm 1,00 trên 1,00

Implement static method selectionSort in class **Sorting** to sort an array in ascending order. After each selection, we will print out a list to check (using printArray).

```
#include <iostream>
using namespace std;

template <class T>
class Sorting
{
public:
    /* Function to print an array */
    static void printArray(T *start, T *end)
    {
        int size = end - start;
        for (int i = 0; i < size - 1; i++)
            cout << start[i] << ", ";
        cout << start[size - 1];
        cout << endl;
    }

    static void selectionSort(T *start, T *end);
};
```

For example:

Test	Result
int arr[] = {9, 2, 8, 1, 0, -2}; Sorting<int>::selectionSort(&arr[0], &arr[6]);	-2, 2, 8, 1, 0, 9 -2, 0, 8, 1, 2, 9 -2, 0, 1, 8, 2, 9 -2, 0, 1, 2, 8, 9 -2, 0, 1, 2, 8, 9

Answer: (penalty regime: 0 %)

Reset answer

```
template <class T>
void Sorting<T>::selectionSort(T *start, T *end) {
    int size = end - start;
    for (int i = 0; i < size - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < size; j++) {
            if (start[j] < start[minIndex]) {
                minIndex = j;
            }
        }
        // Hoán đổi phần tử nhỏ nhất với phần tử đầu tiên của phần chưa sắp xếp
        if (minIndex != i) {
            swap(start[i], start[minIndex]);
        }
        printArray(start, end); // In mảng sau mỗi lần chọn phần tử nhỏ nhất
    }
}
```

	Test	Expected	Got	

✓	`int arr[] = {9, 2, 8, 1, 0, -2};`
	`Sorting<int>::selectionSort(&arr[0], &arr[6]);`
		`-2, 2, 8, 1, 0, 9`
		`-2, 0, 8, 1, 2, 9`
		`-2, 0, 1, 8, 2, 9`
		`-2, 0, 1, 2, 8, 9`
		`-2, 0, 1, 2, 8, 9`
			`-2, 2, 8, 1, 0, 9`
			`-2, 0, 8, 1, 2, 9`
			`-2, 0, 1, 8, 2, 9`
			`-2, 0, 1, 2, 8, 9`
			`-2, 0, 1, 2, 8, 9`
				✓

Passed all tests! ✓

Đúng

Marks for this submission: 1,00/1,00.

Câu hỏi **3**

Sai

Đạt điểm 0,00 trên 1,00

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

```
template <class T>
static void Sorting<T>::sortSegment(T* start, T* end, int segment_idx, int cur_segment_total) {
    int size = end - start;
    for (int i = segment_idx + cur_segment_total; i < size; i += cur_segment_total) {
        T key = start[i];
        int j = i - cur_segment_total;
        while (j >= 0 && start[j] > key) {
            start[j + cur_segment_total] = start[j];
            j -= cur_segment_total;
        }
        start[j + cur_segment_total] = key;
    }
}

template <class T>
static void Sorting<T>::ShellSort(T* start, T* end, int* num_segment_list, int num_phases) {
    for (int i = 0; i < num_phases; i++) {
        int cur_segment_total = num_segment_list[i];
        for (int segment_idx = 0; segment_idx < cur_segment_total; segment_idx++) {
            sortSegment(start, end, segment_idx, cur_segment_total);
        }
        cout << cur_segment_total << " segments: ";
        printArray(start, end);
    }
}
```

Syntax Error(s)

```
__tester__.cpp:25:15: error: declaration of template parameter 'T' shadows template parameter
   25 |     template <class T>
      |               ^~~~
__tester__.cpp:10:11: note: template parameter 'T' declared here
   10 | template <class T>
      |           ^~~~
__tester__.cpp:26:93: error: invalid use of incomplete type 'class Sorting<T>'
   26 | static void Sorting<T>::sortSegment(T* start, T* end, int segment_idx, int cur_segment_total) {
      |                                                                                             ^
__tester__.cpp:11:7: note: declaration of 'class Sorting<T>'
   11 | class Sorting {
      |       ^~~~~~
__tester__.cpp:39:11: error: declaration of template parameter 'T' shadows template parameter
   39 | template <class T>
      |           ^~~~
__tester__.cpp:10:11: note: template parameter 'T' declared here
   10 | template <class T>
      |           ^~~~
__tester__.cpp:40:90: error: invalid use of incomplete type 'class Sorting<T>'
   40 | static void Sorting<T>::ShellSort(T* start, T* end, int* num_segment_list, int num_phases) {
      |                                                                                          ^
__tester__.cpp:11:7: note: declaration of 'class Sorting<T>'
   11 | class Sorting {
      |       ^~~~~~
__tester__.cpp: In function 'int main()':
__tester__.cpp:63:15: error: 'ShellSort' is not a member of 'Sorting<int>'
   63 | Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:71:15: error: 'ShellSort' is not a member of 'Sorting<int>'
   71 | Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:79:15: error: 'ShellSort' is not a member of 'Sorting<int>'
   79 | Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:87:15: error: 'ShellSort' is not a member of 'Sorting<int>'
   87 | Sorting<int>::ShellSort(&array[0], &array[10], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:95:15: error: 'ShellSort' is not a member of 'Sorting<int>'
   95 | Sorting<int>::ShellSort(&array[0], &array[15], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:103:15: error: 'ShellSort' is not a member of 'Sorting<int>'
  103 | Sorting<int>::ShellSort(&array[0], &array[15], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:111:15: error: 'ShellSort' is not a member of 'Sorting<int>'
  111 | Sorting<int>::ShellSort(&array[0], &array[15], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:119:15: error: 'ShellSort' is not a member of 'Sorting<int>'
  119 | Sorting<int>::ShellSort(&array[0], &array[20], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:127:15: error: 'ShellSort' is not a member of 'Sorting<int>'
  127 | Sorting<int>::ShellSort(&array[0], &array[20], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
__tester__.cpp:135:15: error: 'ShellSort' is not a member of 'Sorting<int>'
  135 | Sorting<int>::ShellSort(&array[0], &array[20], &num_segment_list[0], num_phases);;
      |               ^~~~~~~~~
```

Sai

Marks for this submission: 0,00/1,00.