CS 115 - Assignment 5
Dr Malek Mouhoub

Project

Extend the functionalities of the template class `indexList` (available in the course webpage) by including the following functions.

1. `void selSort()` : sorts an indexed list using selection sort algorithm.
2. `int binSearch(const T&) const`: finds (returns) index of a target item using binary search algorithm. Returns -1 if target item is not found.
3. Concatenate two lists (see example below).
4. Merge two sorted lists (see example below).

A sample run follows.

```
List 1 of type int
Enter number of list items to read: 4
Enter next item - 1
Enter next item - 7
Enter next item - 5
Enter next item - 3

List 2 of type int
Enter number of list items to read: 3
Enter next item - 6
Enter next item - 2
Enter next item - 4

List 1 sorted using selection sort
1
3
5
7

List 2 sorted using selection sort
2
4
6

List 1 concatenated to list 2
1
3
5
7
2
4
6
```

```
List 1 merged to list 2
1
2
3
4
5
6
7

Enter item to search in List 1: 5
5 is found at position 2

Enter item to search in List 2: 12
12 is not found

List 1 of type string
Enter number of list items to read: 3
Enter next item - beta
Enter next item - alpha
Enter next item - gamma

List 2 of type string
Enter number of list items to read: 4
Enter next item - omega
Enter next item - epsilon
Enter next item - lamda
Enter next item - delta

List 1 sorted using selection sort
alpha
beta
gamma

List 2 sorted using selection sort
delta
epsilon
lamda
omega

List 1 concatenated to list 2
alpha
beta
gamma
delta
epsilon
lamda
omega

List 1 merged to list 2
alpha
beta
delta
epsilon
gamma
```

```
lamda
omega

Enter item to search in List 1: beta
beta is found at position 1

Enter item to search in List 2: sigma
sigma is not found
```

Hand In

1. The header, implementation and driver program should be respectively named : `indexList.h, indexList.cpp` and `indexListTest.cpp`. 2. Your C++ program **SHOULD** compile using `CC` (Sun compiler) under Hercules. Use the makefile available on the course webpage.
2. Submit all the above files using WebCT : `www.uregina.ca/webct`. You will then receive an acknowledgement email confirming your submission. You should save this email as a proof of submission. If you do not receive an email acknowledging your submission then you should promptly email the marker (mark115@cs.uregina.ca) with your submission in attachment.

Marking scheme 100% + 10% (Bonus)

1. Readability (program style) : 10%
   - Program easy to read,
   - well commented,
   - good structured (layout, indentation, whitespace, . . .) and designed(following the top-down approach)
2. Compiling and execution process : 10%
   - program compiles (with CC under hercules) w/o errors and warnings
   - robustness : execution w/o run time errors
3. Correctness : 80%
   - code produces correct results (output).
   - **output meets the initial requirements (see above for the output format)**.
4. Bonus : 10%
   - Features that increase functionality and/or presentation.