Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Spring 2020

Homework 3 – Credit Card Black List Processing with a Hybrid Linked List Structure

Due: 11/03/2020, Wednesday, 21:00

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!

Introduction

In this homework, you are asked to implement a program that stores blacklisted credit cards organized according to their expiration dates as in your previous homework, but with several differences that makes this one more complicated.

Your program must use a hybrid data structure that contains both *Doubly Linked List* and *Singly Linked Lists.* The expiration dates must be stored in a *Doubly Linked List*, while the credit card numbers are stored in *Singly Linked Lists* connected to the nodes of the doubly linked list according to the expiration dates. Thus, the data structure that you will use is a hybrid and kind of two-dimensional one. Other structural differences between this and the previous homework are: (i) credit card numbers may not be unique that makes insertion and search operations a bit more complex, and (ii) credit cards numbers must be stored in a sorted fashion (smaller to larger). Your program will read data from a file as a menu option. Moreover, your program must be able to perform several other operations that are different than the previous homework as well. We will explain the details in the subsequent sections of this homework specification.

The Program Flow

At the very beginning, the program prompts a menu that contains a list of operations. The menu provides 6 different options and each option can be chosen in any order several times. When an option is entered, the corresponding operation is performed and then the menu is displayed again. This continues until the user chooses the *Exit* option to end the program. You also have to handle the case where a wrong menu option is entered by the user. The menu options and the corresponding operations are explained below.

1. **Upload card(s) from a file:**

In this option, the program gets input data from a text file. First, the program asks the name of the input file. If the file is not opened successfully, then your program should display an error message and return back to the menu.

After successfully opening the file, your program is going to start storing credit card information by reading the file line by line. Each line of the file contains three pieces of information regarding a credit card. The first entry of the line is the 16-digit credit card number. And the second one is the month of the expiration date (to be read as positive integer and must be between and including 1 and 12). The last one is the year (to be read as an integer; any integer could be used as a year value) of the expiration date for this credit card. You can assume the file contains correct inputs, so no input checks are required for the content of the .txt file.

The data will be stored in a data structure. Your program should read each line and modify the data structure by adding the cards to it. Details about the data structure and how to modify it will be given in the upcoming parts of the document.

An example input file is shown in Figure 1.

```
1740948824551711 2 2023
5276142322168576 2 2023
1892795431233411 11 2000
3874277931986502 5 1992
8602486509006138 9 2035
9344606618496378 4 2026
8291359840763615 7 1995
4209737260165754 3 2000
1200146071777733 6 2001
5998182660380125 4 2031
0947835120164061 5 2031
8984143988087783 3 2015
8371073496510996 5 1992
8348499255333743 8 2000
8088068198972282 1 2000
8907815861242586 10 2021
2653924618211976 11 2021
2952003918195325 2 2023
2586772294196982 3 2045
5549125083939679 11 2020
4558796419498964 2 2023
4558796419498965 2 2023
4558796419498963 2 2023
4558796419498963 2 2023
6154563132165845 12 2020
```

Figure 1. Sample input file

2. **Display List Chronological:**

When the user selects this option, your program should display all the credit cards grouped by their expiration dates. This display operation must be sorted with respect to the expiration year and month (earliest to latest). The cards with the same expiration date must be displayed in sorted fashion (smaller to larger). See sample runs for example outputs.

3. Display List Reverse Chronological:

When the user selects this option, your program should display all the credit cards grouped by their expiration dates. This display operation must be in **reverse** order with respect to the expiration year and month (latest to earliest). The cards with the same expiration date must be displayed in sorted fashion (smaller to larger). See sample runs for example outputs.

4. Card Search:

When the user selects this option, your program should ask for a 16-digit credit card number. Although credit card numbers are integers, due to its length you cannot store it as an `int`, so use string there. After reading the credit card number, first you will make an input check to see if it is really 16-digits and contains only digits. If the input is wrong, you have to display an error message and return back to the menu. If the input is correct, your program tries to find whether there is credit card(s) with given number and display an appropriate message (if credit card(s) is/are found, message should include expiration date(s)). Since credit card numbers are not assumed to be unique in this homework, your program may find more than one credit card with the same number but with different expiration dates. In such a case, your program should display all of them with their expiration dates in chronological order (earliest to latest). See sample runs for example outputs.

5. Bulk Delete:

When the user selects this option, your program should ask for a date (month and year). After taking these inputs, first you need to check if the inputs are valid. Entering non-integer is one of the input checks that you have to perform. Moreover, the month input must be between and including 1 and 12. Any integer value is OK for year. If the inputs are **not** correct, your program should display an error message and return back to the menu. If the inputs are correct, your program should delete all the expiration date nodes and their corresponding credit cards, of which the expiration dates are earlier than or equal to the entered expiration date inputs. Your program should display every deleted credit card numbers grouped by their expiration dates. If no credit card is deleted, then your program should display an informative message. Please see the sample runs about these messages and their content.

6. Exit:

When this option is selected, your program is terminated. In order to make sure that you make **no memory leak**, your program **must** return all the dynamically allocated memory to the heap before the termination.

The Data Structure to be Used

In this homework, you must represent expiration dates as a *doubly linked list*, and represent credit card numbers as *singly linked lists* of which the heads are stored in the expiration date nodes. As a result, you are given two different node types: one for the linked list that holds the credit card numbers, and one for the doubly linked list to hold expiration dates.

The node type for the credit card numbers is a regular one for singly linked lists with one `next` pointer and one string for the credit card number. Partial node `struct` for the credit card node is given below.

```
struct creditCardNode
{
	string creditCardNo;
	creditCardNode * next;

	// constructors come here
};
```

The node type to hold expiration date information is the one for the *doubly linked list*. It has one `next` and one `prev` pointers of the same node pointer type. Moreover, it also has a third pointer to point the `head` of the *singly linked list* for the credit cards. In addition, this node type also keeps two integers; one for year, one for month. Partial node `struct` is given below.

```
struct expirationNode
{
	int month, year;
	creditCardNode * cHead;
	expirationNode * next;
	expirationNode * prev;

	// constructors come here
};
```

You are requested to design and implement a class for the *doubly linked list*. Name this class `CardList`, keep the `head` and `tail` as private data members and perform all operations via the member functions of this class (including the operations regarding the credit cards). Partial class definition is given below. You can add some helper functions, if you need to do so. You can copy/paste this class definition and structs given above in a header file. However, you have to implement the member functions in another .cpp file (other than the one includes main) and use them in your main program.

```
class CardList
{
public:
	CardList(); //default constructor that creates an empty list
	void insertCard (string creditCardNo, int month, int year);
		//inserts a new card to the structure in sorted fashion
	void displayListChronological ();
		//displays entire structure in chronological order
	void displayListReverseChronological ();
		//displays entire structure in reverse chronological order
	void cardSearch (string creditCardNo);
		//displays all of the occurrences of the given card number
	void bulkDelete (int month, int year);
		//deletes all nodes up to and including given expiration date
	void deleteAll (); //deletes the entire structure
private:
	expirationNode * head;
	expirationNode * tail;

	// any helper functions you see necessary
};
```

No other multiple data containers (built-in array, dynamic array, vector, matrix, 2D array, etc.) can be used.

Figure 2 depicts the data structure that you will use in this homework. As seen in the figure, a doubly linked list will be used to store the expiration dates data and a singly linked list will be used to store the credit card numbers.

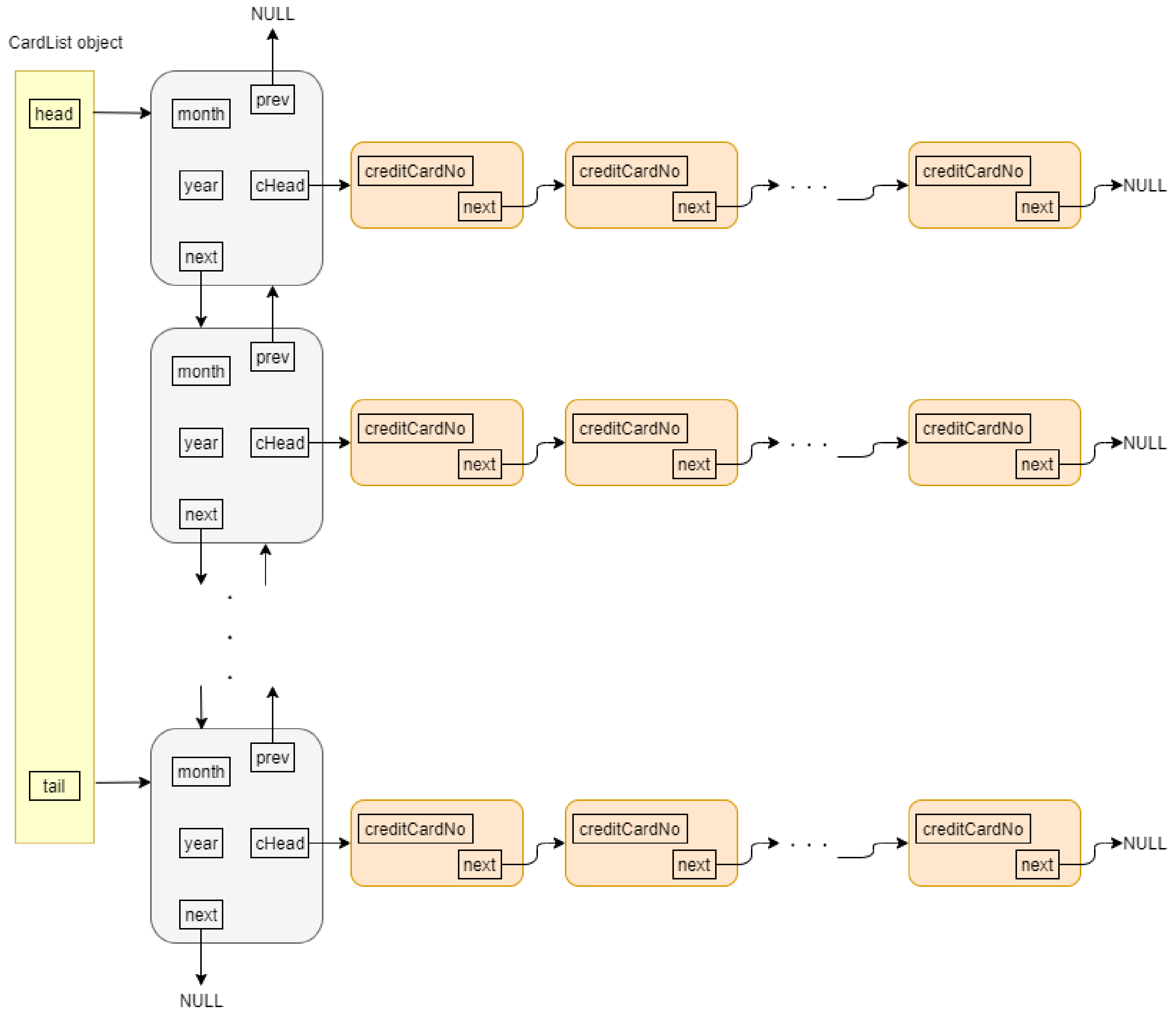

Figure 2. The data structure

Addition and Deletion of Data to the Data Structure

In the hybrid data structure, all the credit cards with the same expiration year and month must be stored as a singly linked list. The head of this list is stored in the corresponding expiration date node. Thus, do not create a new expiration date node for a credit card if there already exists an expiration date node with that card's expiration year and month; instead insert a new credit card node to the credit card list of this expiration date node. However, if there is not a node with new credit card's expiration year and month, then create and insert a new expiration date node and add the new credit card there. This mechanism guarantees that there will not be two nodes with the same expiration year and month in the data structure.

Credit card numbers are not assumed to be unique such that there can be two or more credit cards with the same number but different expiration dates. Such cards must be stored within different expiration date nodes. On the other hand, if a card with the same number and same expiration date is seen more than once, then you should not add multiple copies of the same credit card node; instead you should just display an appropriate message (see sample runs).

After processing each line of the input file and adding the card information to the data structure, you have to display a single line message explaining the steps taken (new expiration node created or added to an existing expiration node or existing credit card node, etc.); please see the sample runs for these messages.

The doubly linked list for the expiration dates **must always be kept** in chronological order (earliest to latest) from head to tail. The credit card lists **must always be kept** in ascending order of credit card numbers (smallest to largest). These are going to be checked in grading, so follow these rules strictly.

Whenever an expiration date node needs to be deleted, you must deallocate all the corresponding credit card nodes connected to it, before deallocating the expiration date node itself.

Moreover, you must deallocate all dynamic memory before your program terminates. You have to perform this by implementing and calling `deleteAll` member function of the class.

Some Important Rules

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate. Since using classes is mandated in this homework, a proper object-oriented design and implementation will also be considered in grading.

Since you will use dynamic memory allocation in this homework, it is very crucial to properly manage the allocated area and return the deleted parts to the heap whenever appropriate. Inefficient use of memory may reduce your grade.

When we grade your homework we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may run your programs in *Release* mode and **we may test your programs with very large test cases**. Of course, your program should work in *Debug* mode as well.

You are allowed to use sample codes shared with the class by the instructor and TAs. However, you cannot start with an existing .cpp or .h file directly and update it; you have start with an empty file. Only the necessary parts of the shared code files can be used and these parts must be clearly marked in your homework by putting comments like the following. Even if you take a piece of code and update it slightly, you have to put a similar marking (by adding "`and updated`" to the comments below.

```
/* Begin: code taken from ptrfunc.cpp */

…

/* End: code taken from ptrfunc.cpp */
```

Sample Runs

Sample runs are given below, but these are not comprehensive, therefore you have to consider **all possible cases** to get full mark.

The sample input files are provided in the .zip package of this homework.

Sample run 1:

```
1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 2

List is empty!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

List is empty!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 1234567890123456

There is no credit card with given credit card number: 1234567890123456

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 11
Invalid operation!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditcards.txt
```

```
Could not find a file named creditcards.txt

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards1.txt

1740948824551711 2 2023: added to a new expiration date node
5276142322168576 2 2023: inserted to an existing expiration date node
1892795431233411 11 2000: added to a new expiration date node
3874277931986502 5 1992: added to a new expiration date node
8602486509006138 9 2035: added to a new expiration date node
4209737260165754 3 2000: added to a new expiration date node
8348499255333743 8 2000: added to a new expiration date node
8088068198972282 1 2000: added to a new expiration date node
4558796419498964 2 2023: inserted to an existing expiration date node
4558796419498965 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: this card was already inserted
6154563132165845 12 2020: added to a new expiration date node
8348499255333743 8 2000: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: ASDQWEASDQWEASDQ
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 12345678901234567
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 123456789012345
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
```

```
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 4558796419498965

There exists a credit card given number 4558796419498965 with expiration date:
2 2023

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: ASD 2010
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: 13 2010
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 ASD
Invalid format!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 2010

Node with expiration date 5 1992  and the following credit cards have been
deleted!
1) 3874277931986502
Node with expiration date 1 2000  and the following credit cards have been
deleted!
1) 8088068198972282
Node with expiration date 3 2000  and the following credit cards have been
deleted!
1) 4209737260165754
Node with expiration date 8 2000  and the following credit cards have been
deleted!
1) 8348499255333743
Node with expiration date 11 2000  and the following credit cards have been
deleted!
1) 1892795431233411
```

```
1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 8088068198972282

There is no credit card with given credit card number: 8088068198972282

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 2

Expiration Date: 12  2020
1) 6154563132165845
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 5276142322168576
-------------------
Expiration Date: 9  2035
1) 8602486509006138
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

Expiration Date: 9  2035
1) 8602486509006138
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 5276142322168576
-------------------
Expiration Date: 12  2020
1) 6154563132165845
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
```

```
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 2100

Node with expiration date 12 2020  and the following credit cards have been
deleted!
1) 6154563132165845
Node with expiration date 2 2023  and the following credit cards have been
deleted!
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 5276142322168576
Node with expiration date 9 2035  and the following credit cards have been
deleted!
1) 8602486509006138

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 2

List is empty!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

List is empty!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 6
All the nodes have been deleted!
Terminating!!!
```

Sample run 2:

```
1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards1.txt

1740948824551711 2 2023: added to a new expiration date node
5276142322168576 2 2023: inserted to an existing expiration date node
```

```
1892795431233411 11 2000: added to a new expiration date node
3874277931986502 5 1992: added to a new expiration date node
8602486509006138 9 2035: added to a new expiration date node
4209737260165754 3 2000: added to a new expiration date node
8348499255333743 8 2000: added to a new expiration date node
8088068198972282 1 2000: added to a new expiration date node
4558796419498964 2 2023: inserted to an existing expiration date node
4558796419498965 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: this card was already inserted
6154563132165845 12 2020: added to a new expiration date node
8348499255333743 8 2000: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 1740948824551711

There exists a credit card given number 1740948824551711 with expiration date:
2 2023

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards2.txt

1740948824551711 2 2021: added to a new expiration date node
5276142322168576 2 2022: added to a new expiration date node
1892795431233411 11 2030: added to a new expiration date node
3874277931986502 5 1991: added to a new expiration date node
8602486509006138 9 2032: added to a new expiration date node
9344606618496378 4 2024: added to a new expiration date node
5998182660380125 4 2031: added to a new expiration date node
0947835120164061 5 2031: added to a new expiration date node
8984143988087783 3 2010: added to a new expiration date node
8371073496510996 5 1992: inserted to an existing expiration date node
8348499255333743 10 2010: added to a new expiration date node
2586772294196982 3 2045: added to a new expiration date node
5549125083939679 3 2045: inserted to an existing expiration date node
4558796419498964 8 2023: added to a new expiration date node
4558796419498965 10 2010: inserted to an existing expiration date node
4558796419498969 2 2023: inserted to an existing expiration date node

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 1740948824551711
```

```
There exists a credit card given number 1740948824551711 with expiration date:
2 2021
There exists a credit card given number 1740948824551711 with expiration date:
2 2023

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 2010

Node with expiration date 5 1991  and the following credit cards have been
deleted!
1) 3874277931986502
Node with expiration date 5 1992  and the following credit cards have been
deleted!
1) 3874277931986502
2) 8371073496510996
Node with expiration date 1 2000  and the following credit cards have been
deleted!
1) 8088068198972282
Node with expiration date 3 2000  and the following credit cards have been
deleted!
1) 4209737260165754
Node with expiration date 8 2000  and the following credit cards have been
deleted!
1) 8348499255333743
Node with expiration date 11 2000  and the following credit cards have been
deleted!
1) 1892795431233411
Node with expiration date 3 2010  and the following credit cards have been
deleted!
1) 8984143988087783
Node with expiration date 10 2010  and the following credit cards have been
deleted!
1) 4558796419498965
2) 8348499255333743

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 2

Expiration Date: 12  2020
1) 6154563132165845
-------------------
Expiration Date: 2  2021
1) 1740948824551711
-------------------
Expiration Date: 2  2022
1) 5276142322168576
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
```

```
3) 4558796419498964
4) 4558796419498965
5) 4558796419498969
6) 5276142322168576
-------------------
Expiration Date: 8  2023
1) 4558796419498964
-------------------
Expiration Date: 4  2024
1) 9344606618496378
-------------------
Expiration Date: 11  2030
1) 1892795431233411
-------------------
Expiration Date: 4  2031
1) 5998182660380125
-------------------
Expiration Date: 5  2031
1) 0947835120164061
-------------------
Expiration Date: 9  2032
1) 8602486509006138
-------------------
Expiration Date: 9  2035
1) 8602486509006138
-------------------
Expiration Date: 3  2045
1) 2586772294196982
2) 5549125083939679
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 2022

Node with expiration date 12 2020  and the following credit cards have been
deleted!
1) 6154563132165845
Node with expiration date 2 2021  and the following credit cards have been
deleted!
1) 1740948824551711
Node with expiration date 2 2022  and the following credit cards have been
deleted!
1) 5276142322168576

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 4
Please enter the credit card number: 1740948824551711

There exists a credit card given number 1740948824551711 with expiration date:
2 2023
```

```
1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards3.txt


1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 6
All the nodes have been deleted!
Terminating!!!
```

Sample Run 3:

```
1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards1.txt

1740948824551711 2 2023: added to a new expiration date node
5276142322168576 2 2023: inserted to an existing expiration date node
1892795431233411 11 2000: added to a new expiration date node
3874277931986502 5 1992: added to a new expiration date node
8602486509006138 9 2035: added to a new expiration date node
4209737260165754 3 2000: added to a new expiration date node
8348499255333743 8 2000: added to a new expiration date node
8088068198972282 1 2000: added to a new expiration date node
4558796419498964 2 2023: inserted to an existing expiration date node
4558796419498965 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: this card was already inserted
6154563132165845 12 2020: added to a new expiration date node
8348499255333743 8 2000: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards1.txt

1740948824551711 2 2023: this card was already inserted
5276142322168576 2 2023: this card was already inserted
1892795431233411 11 2000: this card was already inserted
```

```
3874277931986502 5 1992: this card was already inserted
8602486509006138 9 2035: this card was already inserted
4209737260165754 3 2000: this card was already inserted
8348499255333743 8 2000: this card was already inserted
8088068198972282 1 2000: this card was already inserted
4558796419498964 2 2023: this card was already inserted
4558796419498965 2 2023: this card was already inserted
4558796419498963 2 2023: this card was already inserted
4558796419498963 2 2023: this card was already inserted
6154563132165845 12 2020: this card was already inserted
8348499255333743 8 2000: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

Expiration Date: 9  2035
1) 8602486509006138
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 5276142322168576
-------------------
Expiration Date: 12  2020
1) 6154563132165845
-------------------
Expiration Date: 11  2000
1) 1892795431233411
-------------------
Expiration Date: 8  2000
1) 8348499255333743
-------------------
Expiration Date: 3  2000
1) 4209737260165754
-------------------
Expiration Date: 1  2000
1) 8088068198972282
-------------------
Expiration Date: 5  1992
1) 3874277931986502
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards2.txt

1740948824551711 2 2021: added to a new expiration date node
5276142322168576 2 2022: added to a new expiration date node
1892795431233411 11 2030: added to a new expiration date node
```

```
3874277931986502 5 1991: added to a new expiration date node
8602486509006138 9 2032: added to a new expiration date node
9344606618496378 4 2024: added to a new expiration date node
5998182660380125 4 2031: added to a new expiration date node
0947835120164061 5 2031: added to a new expiration date node
8984143988087783 3 2010: added to a new expiration date node
8371073496510996 5 1992: inserted to an existing expiration date node
8348499255333743 10 2010: added to a new expiration date node
2586772294196982 3 2045: added to a new expiration date node
5549125083939679 3 2045: inserted to an existing expiration date node
4558796419498964 8 2023: added to a new expiration date node
4558796419498965 10 2010: inserted to an existing expiration date node
4558796419498969 2 2023: inserted to an existing expiration date node

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards2.txt

1740948824551711 2 2021: this card was already inserted
5276142322168576 2 2022: this card was already inserted
1892795431233411 11 2030: this card was already inserted
3874277931986502 5 1991: this card was already inserted
8602486509006138 9 2032: this card was already inserted
9344606618496378 4 2024: this card was already inserted
5998182660380125 4 2031: this card was already inserted
0947835120164061 5 2031: this card was already inserted
8984143988087783 3 2010: this card was already inserted
8371073496510996 5 1992: this card was already inserted
8348499255333743 10 2010: this card was already inserted
2586772294196982 3 2045: this card was already inserted
5549125083939679 3 2045: this card was already inserted
4558796419498964 8 2023: this card was already inserted
4558796419498965 10 2010: this card was already inserted
4558796419498969 2 2023: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

Expiration Date: 3  2045
1) 2586772294196982
2) 5549125083939679
-------------------
Expiration Date: 9  2035
1) 8602486509006138
-------------------
Expiration Date: 9  2032
1) 8602486509006138
-------------------
Expiration Date: 5  2031
1) 0947835120164061
-------------------
```

```
Expiration Date: 4  2031
1) 5998182660380125
-------------------
Expiration Date: 11  2030
1) 1892795431233411
-------------------
Expiration Date: 4  2024
1) 9344606618496378
-------------------
Expiration Date: 8  2023
1) 4558796419498964
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 4558796419498969
6) 5276142322168576
-------------------
Expiration Date: 2  2022
1) 5276142322168576
-------------------
Expiration Date: 2  2021
1) 1740948824551711
-------------------
Expiration Date: 12  2020
1) 6154563132165845
-------------------
Expiration Date: 10  2010
1) 4558796419498965
2) 8348499255333743
-------------------
Expiration Date: 3  2010
1) 8984143988087783
-------------------
Expiration Date: 11  2000
1) 1892795431233411
-------------------
Expiration Date: 8  2000
1) 8348499255333743
-------------------
Expiration Date: 3  2000
1) 4209737260165754
-------------------
Expiration Date: 1  2000
1) 8088068198972282
-------------------
Expiration Date: 5  1992
1) 3874277931986502
2) 8371073496510996
-------------------
Expiration Date: 5  1991
1) 3874277931986502
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
```

```
6) Exit
Please choose option from the menu: 5
Please enter month and year: 10 2100

Node with expiration date 5 1991  and the following credit cards have been
deleted!
1) 3874277931986502
Node with expiration date 5 1992  and the following credit cards have been
deleted!
1) 3874277931986502
2) 8371073496510996
Node with expiration date 1 2000  and the following credit cards have been
deleted!
1) 8088068198972282
Node with expiration date 3 2000  and the following credit cards have been
deleted!
1) 4209737260165754
Node with expiration date 8 2000  and the following credit cards have been
deleted!
1) 8348499255333743
Node with expiration date 11 2000  and the following credit cards have been
deleted!
1) 1892795431233411
Node with expiration date 3 2010  and the following credit cards have been
deleted!
1) 8984143988087783
Node with expiration date 10 2010  and the following credit cards have been
deleted!
1) 4558796419498965
2) 8348499255333743
Node with expiration date 12 2020  and the following credit cards have been
deleted!
1) 6154563132165845
Node with expiration date 2 2021  and the following credit cards have been
deleted!
1) 1740948824551711
Node with expiration date 2 2022  and the following credit cards have been
deleted!
1) 5276142322168576
Node with expiration date 2 2023  and the following credit cards have been
deleted!
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
4) 4558796419498965
5) 4558796419498969
6) 5276142322168576
Node with expiration date 8 2023  and the following credit cards have been
deleted!
1) 4558796419498964
Node with expiration date 4 2024  and the following credit cards have been
deleted!
1) 9344606618496378
Node with expiration date 11 2030  and the following credit cards have been
deleted!
1) 1892795431233411
Node with expiration date 4 2031  and the following credit cards have been
deleted!
1) 5998182660380125
Node with expiration date 5 2031  and the following credit cards have been
deleted!
```

```
1) 0947835120164061
Node with expiration date 9 2032  and the following credit cards have been
deleted!
1) 8602486509006138
Node with expiration date 9 2035  and the following credit cards have been
deleted!
1) 8602486509006138
Node with expiration date 3 2045  and the following credit cards have been
deleted!
1) 2586772294196982
2) 5549125083939679

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

List is empty!

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 1
Please enter file name: creditCards1.txt

1740948824551711 2 2023: added to a new expiration date node
5276142322168576 2 2023: inserted to an existing expiration date node
1892795431233411 11 2000: added to a new expiration date node
3874277931986502 5 1992: added to a new expiration date node
8602486509006138 9 2035: added to a new expiration date node
4209737260165754 3 2000: added to a new expiration date node
8348499255333743 8 2000: added to a new expiration date node
8088068198972282 1 2000: added to a new expiration date node
4558796419498964 2 2023: inserted to an existing expiration date node
4558796419498965 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: inserted to an existing expiration date node
4558796419498963 2 2023: this card was already inserted
6154563132165845 12 2020: added to a new expiration date node
8348499255333743 8 2000: this card was already inserted

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 3

Expiration Date: 9  2035
1) 8602486509006138
-------------------
Expiration Date: 2  2023
1) 1740948824551711
2) 4558796419498963
3) 4558796419498964
```

```
4) 4558796419498965
5) 5276142322168576
-------------------
Expiration Date: 12  2020
1) 6154563132165845
-------------------
Expiration Date: 11  2000
1) 1892795431233411
-------------------
Expiration Date: 8  2000
1) 8348499255333743
-------------------
Expiration Date: 3  2000
1) 4209737260165754
-------------------
Expiration Date: 1  2000
1) 8088068198972282
-------------------
Expiration Date: 5  1992
1) 3874277931986502
-------------------

1) Upload Card(s) from a File
2) Display List Chronological
3) Display List Reverse Chronological
4) Card Search
5) Bulk Delete
6) Exit
Please choose option from the menu: 6
All the nodes have been deleted!
Terminating!!!
```

What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process might be automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Name your cpp file that contains your main program using the following convention:

“SUCourseUserName_YourLastname_YourName_HWnumber.cpp”

Your SUCourse user name is your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw3.cpp

In some homework assignments, you may need to have more than one .cpp or .h files to submit. In this case, add informative phrases after the hw number. However, do not add any other character or phrase to the file names. Sometimes, you may want to use some user defined libraries (such as strutils of Tapestry); in such cases, you have to provide the necessary .cpp and .h files of them as well. If you use standard C++ libraries, you do not need to provide extra files for them.

These source files are the ones that you are going to submit as your homework. However, even if you have a single file to submit, you have to compress it using ZIP format. To do so, first create a folder that follows the abovementioned naming convention (“SUCourseUserName_YourLastname_YourName_HWnumber"). Then, copy your source file(s) there. And finally compress this folder using WINZIP or WINRAR programs (or another mechanism). Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains all of the files that belong to the latest version of your homework.

You will receive zero if your compressed zip file does not expand or it does not contain the correct files. The naming convention of the zip file is the same. The name of the zip file should be as follows:

SUCourseUserName_YourLastname_YourName_HWnumber.zip

For example, zubzipler_Zipleroglu_Zubeyir_hw3.zip is a valid name, but

Hw3_hoz_HasanOz.zip, HasanOzHoz.zip

are **NOT** valid names.

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

Albert Levi, Vedat Peran