Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Spring 2020

Homework 2 – Credit Card Black List Processing with Linked Lists

Due: 26/2/2020, 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 and homework trading 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. This program must use a *Linked List* structure for storage. Credit card numbers and their expiration dates are going to be read from a text file. The program details will be explained in the subsequent sections.

The Data Structure to be Used

In this homework, you **must** use a *linked list* (regular one-way linked list) as your main data structure. One *node* struct of this list must have the following data members (if you want, you can add constructors to the struct).

```
struct node
{
     int month, year;
     vector<string> cards;
     node* next;
};
```

In this node structure, `month` and `year` represent the expiration date of the blacklisted cards that are stored in this node and the `cards` vector is the list of the card numbers. The card numbers, although they are of type string, are actually used as 16-digit numbers. You can visualize this data structure in Figure 1.

You are not allowed to use arrays, extra vectors and similar containers (including extra files) in this homework; all data must be stored and processed within the linked list. The only vector that you can use is the one in the node struct.

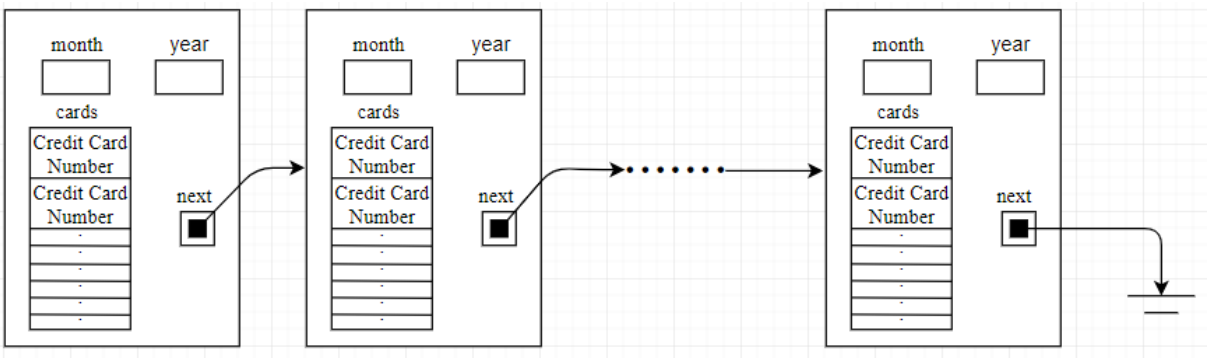

Figure 1: Visualization of Linked List structure

The Program Flow

Your program is going to start with getting an input from the user regarding the name of the .txt file that contains credit card information. After getting the name of the file, the program must check whether the file has opened correctly. If not, another file name will be required from the user until a correct file name is entered.

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 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. Moreover, also assume that the credit card numbers are unique and you do not need to make a check for uniqueness. An example input file is shown in Figure 2.

```
1740948824551711 2 2023
5276142322168576 2 2000
1892795431233411 11 2000
3874277931986502 5 1992
8602486509006138 9 2035
9344606618496378 4 2026
8291359840763615 7 1995
4209737260165754 2 2000
1200146071777733 6 2001
5998182660380125 4 2031
0947835120164061 4 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 12 2020
4558796419498964 2 2000
4464649844541448 2 2000
```

Figure 2: Example input file

In the data structure, all the credit cards with the same expiration year and month must be stored in the same node. That is why we use a vector for cards there. Thus, do not create a new node for a credit card if there already exists a node with that card's expiration year and month; instead push it to the cards vector of that node. However, if there is not a node with new credit card's expiration year and month, then create a node and push the credit card number as the first entry of that node's cards vector. This mechanism guarantees that (i) there will not be a node with an empty cards vector, and (ii) there will not be two nodes with the same expiration year and month in the data structure. This is going to be checked in grading, so follow these rules strictly. After processing each line of the input file and adding the card information to the linked list, you have display output regarding the steps taken (new node created or added to an existing node); please see the sample rums for these messages.

Moreover, your program should maintain the linked list in a sorted fashion with respect to expiration dates (earliest to latest). That means, the head of the list must point to the node with the earliest year and month combination of the list, and the last node's year and month combination must be the latest one. While inserting a new node, you have to preserve the order of the list all the time. This is also going to be checked in grading.

After storing the data read from .txt file into your linked list, your program should display a menu with four different options as given below.

1. Display List
2. Card Search
3. Delete Cards (with respect to Expiration Date)
4. Exit

According to the option value (1, 2, 3, or 4) that the user selects, the system performs the associated operation. Your program reads the option from the standard input (keyboard). If a wrong option is selected, an error message must be displayed and new option must be read.

The operations are explained below.

1. **Display List**
   When 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 the expiration year and month (earliest to latest as stored in the linked list). The cards with the same expiration date must be displayed in the order that they are pushed onto the corresponding cards vector. See sample runs for example outputs.

2. **Card Search**
   When 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 (as you do in the linked list). 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 ask for another card number. After reading a valid credit card number from the user, your program tries to find whether there is a credit card with given number and display an appropriate message (if credit card is found, message should include expiration date). See sample runs for example outputs.

3. **Delete Cards (with respect to Expiration Date)**
   When 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. You have to continue to enter month and year inputs until correct ones are entered. Once they are entered correctly, your program should delete the node, which contains the entered month and year information, from your linked list, if there is such a node. After the deletion, your list must remain sorted. You have to display appropriate output after deletion. If there is no such node with given month and year, then no deletion will be performed, but a message explaining this situation must be displayed. Please see the sample runs about these messages and their content.

4. **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.

After a menu option is selected and the required processing is performed, the menu should be displayed and a new option is selected continuously until the user enters "4" to Exit.

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 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:

```
Please enter file name: creditcards1
Cannot find a file named creditcards1
Please enter file name: creditCards1.docx
Cannot find a file named creditCards1.docx
Please enter file name: creditCards1.txt
New node is created with expiration date: 2 2023
Credit card 1740948824551711 added to node 2 2023
***************
New node is created with expiration date: 2 2000
Credit card 5276142322168576 added to node 2 2000
***************
New node is created with expiration date: 11 2000
Credit card 1892795431233411 added to node 11 2000
***************
New node is created with expiration date: 5 1992
Credit card 3874277931986502 added to node 5 1992
***************
New node is created with expiration date: 9 2035
Credit card 8602486509006138 added to node 9 2035
***************
New node is created with expiration date: 4 2026
Credit card 9344606618496378 added to node 4 2026
***************
New node is created with expiration date: 7 1995
Credit card 8291359840763615 added to node 7 1995
***************
Node with expiration date 2 2000 already exists
Credit card 4209737260165754 added to node 2 2000
***************
New node is created with expiration date: 6 2001
Credit card 1200146071777733 added to node 6 2001
***************
New node is created with expiration date: 4 2031
Credit card 5998182660380125 added to node 4 2031
***************
Node with expiration date 4 2031 already exists
Credit card 0947835120164061 added to node 4 2031
***************
New node is created with expiration date: 3 2015
Credit card 8984143988087783 added to node 3 2015
***************
Node with expiration date 5 1992 already exists
Credit card 8371073496510996 added to node 5 1992
```

```
***************
New node is created with expiration date: 8 2000
Credit card 8348499255333743 added to node 8 2000
***************
New node is created with expiration date: 1 2000
Credit card 8088068198972282 added to node 1 2000
***************
New node is created with expiration date: 10 2021
Credit card 8907815861242586 added to node 10 2021
***************
New node is created with expiration date: 11 2021
Credit card 2653924618211976 added to node 11 2021
***************
Node with expiration date 2 2023 already exists
Credit card 2952003918195325 added to node 2 2023
***************
New node is created with expiration date: 3 2045
Credit card 2586772294196982 added to node 3 2045
***************
New node is created with expiration date: 12 2020
Credit card 5549125083939679 added to node 12 2020
***************
Node with expiration date 2 2000 already exists
Credit card 4558796419498964 added to node 2 2000
***************

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 5  1992
1) 3874277931986502
2) 8371073496510996
-------------------

Expiration Date: 7  1995
1) 8291359840763615
-------------------

Expiration Date: 1  2000
1) 8088068198972282
-------------------

Expiration Date: 2  2000
1) 5276142322168576
2) 4209737260165754
3) 4558796419498964
-------------------
```

Expiration Date: 8 2000
1) 8348499255333743

Expiration Date: 11 2000
1) 1892795431233411

Expiration Date: 6 2001
1) 1200146071777733

Expiration Date: 3 2015
1) 8984143988087783

Expiration Date: 12 2020
1) 5549125083939679

Expiration Date: 10 2021
1) 8907815861242586

Expiration Date: 11 2021
1) 2653924618211976

Expiration Date: 2 2023
1) 1740948824551711
2) 2952003918195325

Expiration Date: 4 2026
1) 9344606618496378

Expiration Date: 4 2031
1) 5998182660380125
2) 0947835120164061

Expiration Date: 9 2035
1) 8602486509006138

Expiration Date: 3 2045
1) 2586772294196982
---

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***2***

Please enter the credit card number: ***ASD***
Invalid format!
Please enter the credit card number: ***123***
Invalid format!
Please enter the credit card number: ***5998182660380125X***
Invalid format!
Please enter the credit card number: ***X5998182660380125***
Invalid format!
Please enter the credit card number: ***XX998182660380125***
Invalid format!
Please enter the credit card number: ***59981826603801255***
Invalid format!
Please enter the credit card number: ***599818266038012***
Invalid format!
Please enter the credit card number: ***5998182660380125***
There exists a credit card given number 5998182660380125 with expiration date: 4 2031

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***2***

Please enter the credit card number: ***1234567890123456***
There is no credit card with given credit card number: 1234567890123456

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***3***

Please enter month and year: ***10 2099***

There is no node with expiration date 10 2099, nothing deleted!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date

```
4)Exit

Please choose option from the menu: 3

Please enter month and year: 10 ASDF
Invalid Date!
Please enter month and year: ASDF 2002
Invalid Date!
Please enter month and year: AS 2002
Invalid Date!
Please enter month and year: 99 ASDF
Invalid Date!
Please enter month and year: 13 2020
Invalid Date!
Please enter month and year: 4 2031
Node with expiration date 4 2031 and the following credit cards
have been deleted!
1)5998182660380125
2)0947835120164061


1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 5  1992
1) 3874277931986502
2) 8371073496510996
-------------------

Expiration Date: 7  1995
1) 8291359840763615
-------------------

Expiration Date: 1  2000
1) 8088068198972282
-------------------

Expiration Date: 2  2000
1) 5276142322168576
2) 4209737260165754
3) 4558796419498964
-------------------

Expiration Date: 8  2000
1) 8348499255333743
-------------------
```

```
Expiration Date: 11  2000
1) 1892795431233411
-------------------

Expiration Date: 6  2001
1) 1200146071777733
-------------------

Expiration Date: 3  2015
1) 8984143988087783
-------------------

Expiration Date: 12  2020
1) 5549125083939679
-------------------

Expiration Date: 10  2021
1) 8907815861242586
-------------------

Expiration Date: 11  2021
1) 2653924618211976
-------------------

Expiration Date: 2  2023
1) 1740948824551711
2) 2952003918195325
-------------------

Expiration Date: 4  2026
1) 9344606618496378
-------------------

Expiration Date: 9  2035
1) 8602486509006138
-------------------

Expiration Date: 3  2045
1) 2586772294196982
-------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3

Please enter month and year: 5 1992
Node with expiration date 5 1992 and the following credit cards
have been deleted!
```

1)3874277931986502
2)8371073496510996

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***1***

Expiration Date: 7 1995
1) 8291359840763615

Expiration Date: 1 2000
1) 8088068198972282

Expiration Date: 2 2000
1) 5276142322168576
2) 4209737260165754
3) 4558796419498964

Expiration Date: 8 2000
1) 8348499255333743

Expiration Date: 11 2000
1) 1892795431233411

Expiration Date: 6 2001
1) 1200146071777733

Expiration Date: 3 2015
1) 8984143988087783

Expiration Date: 12 2020
1) 5549125083939679

Expiration Date: 10 2021
1) 8907815861242586

Expiration Date: 11 2021
1) 2653924618211976

```
-------------------

Expiration Date: 2  2023
1) 1740948824551711
2) 2952003918195325
-------------------

Expiration Date: 4  2026
1) 9344606618496378
-------------------

Expiration Date: 9  2035
1) 8602486509006138
-------------------

Expiration Date: 3  2045
1) 2586772294196982
-------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3

Please enter month and year: 3 2045
Node with expiration date 3 2045 and the following credit cards
have been deleted!
1)2586772294196982


1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 7  1995
1) 8291359840763615
-------------------

Expiration Date: 1  2000
1) 8088068198972282
-------------------

Expiration Date: 2  2000
1) 5276142322168576
2) 4209737260165754
3) 4558796419498964
```

```
-------------------

Expiration Date: 8  2000
1) 8348499255333743
-------------------

Expiration Date: 11  2000
1) 1892795431233411
-------------------

Expiration Date: 6  2001
1) 1200146071777733
-------------------

Expiration Date: 3  2015
1) 8984143988087783
-------------------

Expiration Date: 12  2020
1) 5549125083939679
-------------------

Expiration Date: 10  2021
1) 8907815861242586
-------------------

Expiration Date: 11  2021
1) 2653924618211976
-------------------

Expiration Date: 2  2023
1) 1740948824551711
2) 2952003918195325
-------------------

Expiration Date: 4  2026
1) 9344606618496378
-------------------

Expiration Date: 9  2035
1) 8602486509006138
-------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 4
Terminating!!!
```

Sample Run 2:

```
Please enter file name: creditCards2.txt
New node is created with expiration date: 11 2020
Credit card 9990948824551711 added to node 11 2020
***************
New node is created with expiration date: 2 2000
Credit card 5276142322168576 added to node 2 2000
***************
Node with expiration date 11 2020 already exists
Credit card 1892795431233411 added to node 11 2020
***************

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 2  2000
1) 5276142322168576
--------------------

Expiration Date: 11  2020
1) 9990948824551711
2) 1892795431233411
--------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3

Please enter month and year: 13 2002
Invalid Date!
Please enter month and year: 12 2002

There is no node with expiration date 12 2002, nothing deleted!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 2  2000
1) 5276142322168576
```

```
-------------------

Expiration Date: 11  2020
1) 9990948824551711
2) 1892795431233411
-------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 2

Please enter the credit card number: 1892795431233411
There exists a credit card given number 1892795431233411 with
expiration date: 11 2020


1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3

Please enter month and year: 2 2000
Node with expiration date 2 2000 and the following credit cards
have been deleted!
1)5276142322168576


1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

Expiration Date: 11  2020
1) 9990948824551711
2) 1892795431233411
-------------------

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3
```

Please enter month and year: ***11 2020***
Node with expiration date 11 2020 and the following credit cards have been deleted!
1)9990948824551711
2)1892795431233411

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***1***

List is empty!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***3***

Please enter month and year: ***11 2020***

There is no node with expiration date 11 2020, nothing deleted!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***2***

Please enter the credit card number: ***1892795431233411***
There is no credit card with given credit card number: 1892795431233411

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: ***1***

List is empty!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

```
Please choose option from the menu: 4
Terminating!!!
```

Sample Run 3 (*creditCards3.txt file is an empty file*):

```
Please enter file name: creditCards3.txt

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

List is empty!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 2

Please enter the credit card number: 1234567890123456
There is no credit card with given credit card number:
1234567890123456

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 2

Please enter the credit card number: asdasdasdasdasdasd
Invalid format!
Please enter the credit card number: 12345678901234567
Invalid format!
Please enter the credit card number: 1234567890123456
There is no credit card with given credit card number:
1234567890123456

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

List is empty!
```

```
1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 3

Please enter month and year: 11 2002

There is no node with expiration date 11 2002, nothing deleted!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 1

List is empty!

1)Display List
2)Card Search via Credit Number
3)Delete Card with respect to Expiration Date
4)Exit

Please choose option from the menu: 4
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_hw2.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_hw2.zip is a valid name, but

Hw2_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