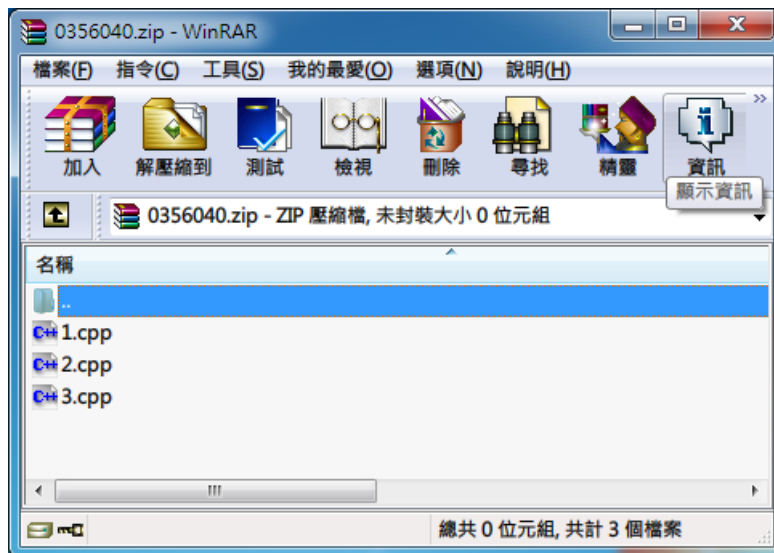


HOMEWORK ASSIGNMENT 2

Due Date: **23:59**, April 2, 2015

Important Notes

Please archive your homework programs as a single zip file and submit the zipped file to E3. The zipped file should be named after your student ID (e.g. 0316000.zip) and includes only 1.cpp, 2.cpp, and 3.cpp for your solutions to Problems 1, 2, and 3, respectively. **Do not change any filenames or add any additional file or directories to the zipped file.** The penalty for incorrectly formatted submission is 10%. Make sure your zipped file match the format as follows:



Note that the penalty for late homework is **15% per day**, and late homework will not be accepted after 3 days past the due date. In addition, homework assignments must be individual work. While you are allowed (and encouraged) to work together in understanding the concepts of the course and even the assigned problems, the solutions that you hand in should be entirely your own. **Sharing of algorithms or code is NOT ALLOWED.** If I detect what I consider to be intentional plagiarism in any assignment, the assignment will receive **zero credit**.

Your programs should **not** call `system("pause")`.

Problem 1

Translate a decimal number to hexadecimal and octal. Your program should keep prompting for a number until EOF is reached.

Hint: you can use the functions or manipulators in `<iostream>` and `<iomanip>`.

Sample Input and Output

```
Please enter a decimal number: 10
Its hexadecimal format is: a
Its octal format is: 12
Please enter a decimal number: ^D
```

Problem 2

Use stream error states to check whether the input is a number or not. Your program should keep prompting for a number until EOF is reached.

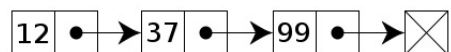
Note: you must use member functions of `istream`.

Sample Input and Output

```
Please enter a number: 3
A number.
Please enter a number: A
Not a number.
Please enter a number: ^D
```

Problem 3

A linked list is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of a data item and a pointer (in other words, a link) to the next node in the sequence, as shown below.



In this question, you need to maintain an ordered linked list under **insert**, **traverse**, **delete**, **unique** and **slice** operations.

Note: you must use `struct node` (listed below) to implement the linked list.

```
struct node{
    int value;
    node *next;
};
```

Insert

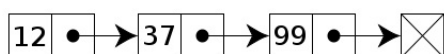
Insert a new node into your linked list. The nodes in the linked list must be linked in ascendant order of node values. If there is no node in your linked list, the newly-allocated node acts as the first node. If a node with the same value exists, the new node can be inserted either before or after the existing node.

Example

Given a linked list:



After inserting 37 into the linked list, the linked list should be:

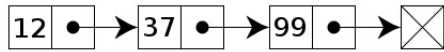


Traverse

Output the values of all nodes from the first to the last node in your linked list.

Example

Suppose we have a linked list:



The traverse function should output: 12 37 99

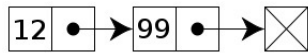
Delete

Given a number as a key, find out which node has the same value and delete the node. If multiple nodes are found, only one of them needs to be removed.

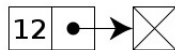
If the node does not exist, print “Node with value [key] is not found!”.

Example

Given the following linked list:



The linked list should be the following after deleting 99:

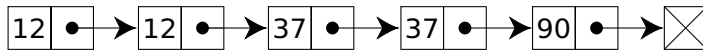


Unique

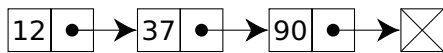
Remove consecutive identical nodes so that the result list contains only unique elements.

Example

Before applying unique function:



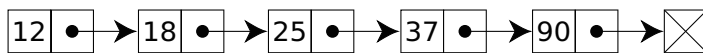
After applying unique function:



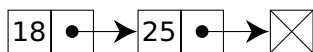
Slice

Given a range $[A, B]$, keep only nodes that satisfy $A \leq V \leq B$, where V is the value of the node.

Example



After slicing with $[15, 25]$ as the range:



Input/Output Format

You have to prompt an action list for users to input which action to perform. The format is as follows:

(1)Insert (2)Traverse (3)Delete (4)Unique (5)Slice (6)Exit:

You shall read a single number between 1 and 6 and perform the corresponding function.

Insert

Insert a value into the linked list with the prompt:

Enter the value of the new node:

Traverse

Print Traverse: followed by the value of each node.

Delete

Ask the value to be deleted with the prompt

Enter the value to be deleted:

Output Node with value [val] is not found! if a node with value [val] is not found in the list.

Unique

Perform the unique function on the linked list. There is no output for this function.

Slice

Prompt with

Enter the range for slicing:

followed by two values (separated by spaces) which represent A and B respectively.

Exit

Terminate the program.

Formatting Note

After the completion of the action, a blank line is printed followed by another prompt message.

Sample Input and Output

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 1
Enter the value of the new node: 10
```

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 1
Enter the value of the new node: 20
```

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 2
Traverse: 10 20
```

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 3
Enter the value to be deleted: 15
Node with value 15 is not found!
```

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 1
Enter the value of the new node: 15
```

```
(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 2
```

Traverse: 10 15 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 3
Enter the value to be deleted: 15

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 1
Enter the value of the new node: 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 1
Enter the value of the new node: 15

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 2
Traverse: 10 15 20 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 4

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 2
Traverse: 10 15 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 5
Enter the range for slicing: 11 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 2
Traverse: 15 20

(1)Insert (2)Traverse (3>Delete (4)Unique (5)Slice (6)Exit: 6