1 Purpose

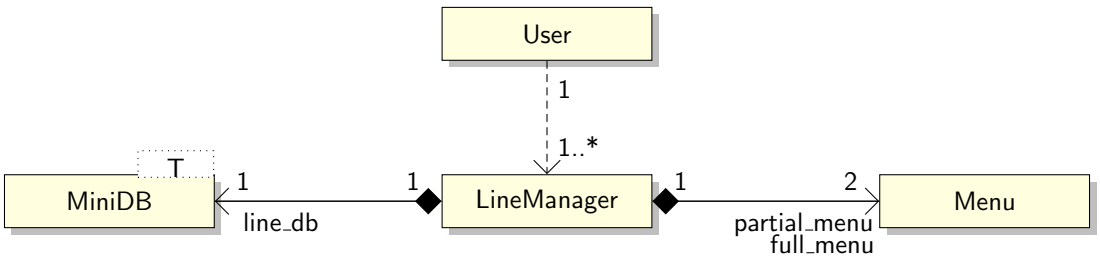
- Avoid explicit use of `new` and `delete`
- Delegate runtime storage management to the Standard Template Library (STL) containers
- Gian experience with using the STL sequential containers
- Use iterators to access and manipulate container elements
- Use code reuse to model "has-a" relationship between classes
- Practice text input/output file processing
- Write a C++ class templates

2 Your Task

A text-based user interface (TUI) presents information textually, on a text-based screen and commands are typed by the user, as opposed to GUI, graphical user interface, where information is presented graphically, typically with drop-down menus, buttons, icons, etc.

Your task in this assignment is to implement a software system that uses an interactive menu-driven TUI to allow the user to create and manage a list of text lines entered via keyboard and/or loaded from text files.

Specifically, your task involves implementing three classes `LineManager`, `Menu`, and `MiniDB` that are related to one another according to the following UML class diagram:



Using the notation table on page 13, the relationship between these classes can be described as follows:

- `MiniDB` is a class template with a generic type parameter `T`.
- An object of `LineManager` is composed of one instance of `MiniDB`, giving it the role name *line_db*.

- An object of `LineManager` is composed of two instances of `Menu`, giving them the role names *partial_menu* and *full_menu*.
- Objects of `Menu` and `MiniDB` have no knowledge of `LineManager` and of each other.
- An object of `User` uses one or more objects of `LineManager`.

An object of `LineManager` acts as an intermediary between the user and an object of type `MiniDB<std::string>` (which in turn models a mini-database), providing interactive text-based services that include

- creating a database of text lines,
- inserting and deleting a text line,
- printing a range of lines,
- loading lines from a text file,
- writing the entire line database to a text file, and

Clearly, for `LineManager` to be effective in what it does, it must be knowledgeable about the functionality of its own data member objects of `Menu` and `MiniDB` and about interaction with the user.

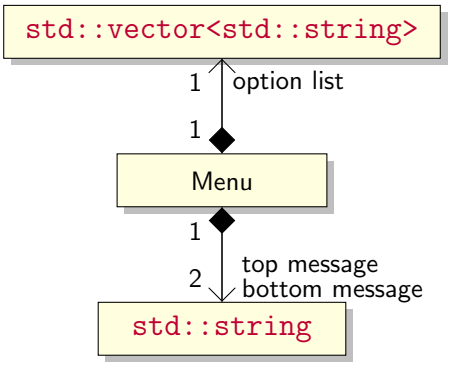
3 Class Menu

An object of class `Menu` supports the same set of operations as those of the `Menu` class in assignment 1, but it uses a different representation for its underlying storage; specifically, it replaces class `Text` with `std::string` and the option list with `std:vector<std::string>`.

Heads up Please note that the `Menu` class in this assignment uses no `Text` objects, no arrays of `Text` objects, no doubling capacity, no `new` and `delete`, no `count` and `capacity` members, and no special member functions (copy/move constructors, copy/move assignment operator overloads, and destructor).

3.1 Representation

- `std:vector<std::string>`, representing the option list
- `std::string`, representing the top and bottom messages



3.2 Operations

See assignment 1, page 10.

4 Class MiniDB

An object of class `MiniDB` models a sequence container that provides basic but efficient database operations, such as insertion, deletion, etc.

What sets `MiniDB` apart from other sequence container classes such as `std::array`, `std::vector`, `std::deque`, `std::list`, and `std::forward_list` is the concept of a **current position** (**cp**) that it supports for its items; the concept is similar to that of the positions *front* and *rear* in a *queue* and that of *top* in a *stack* container, where the primary activities of the containers are localized at those positions, respectively.

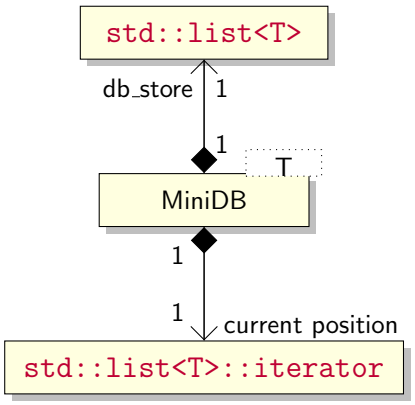
The **cp** maintained by a `MiniDB` object represents the "position" of an item that can be accessed and modified by that object. The goal of this design is to give you plenty of practice with C++ iterators, specifically, the `std::list<T>::iterator` which implements the concept of **cp** in our mini database. Although **cp** is not a pointer, we may say "**cp** points at" rather than "**cp** represents the position of" an item in the database.

Since the operations of our mini-database `MiniDB` are independent of the type of the items in it, we generalize it into a class template `MiniDB<T>` of one type parameter `T` which represents the item type. The class template `MiniDB<T>` will allow the user to "instantiate" real classes such as `MiniDB<int>`, `MiniDB<Foo>`, etc. Our `LineManager` class uses an object of `MiniDB<std::string>` as its underlying database. The goal here is to facilitate your understanding of the STL topics we will cover later on in the course.

4.1 Representation

Given that we want `MiniDB<T>` to model a sequence container, we need to decide what type of storage we want to use to store the items in the database. To do that, we argue as follows:

- We consider using only the highly optimized standard sequence containers.
- We rule `std::array` out because we can't predict the size of the database at compile time.
- We want to move **cp** efficiently both forward and backward within the container; that rules out `std::forward_list`.
- Both `std::vector` and `std::deque` are ruled out because we want efficient insertion and deletion of items everywhere, and not just at one or two ends of the container.
- We are left with `std::list` as the most suitable underlying container for our mini-database.



The storage representation for a `miniDB<T>` object is specified by two data members:

- A `std::list<T>` object called say, `db_store`

- A `std::list<T>::iterator` object called, say, `current_position`

4.2 Operations

4.2.1 Public Interface

1. **Special member functions**
   All defaulted, leaving the responsibility of copy/assignment operations to the corresponding special member functions of the data member objects `db_store` and `current_position`.

2. **size_t size() const**
   Returns the number of items in the database

3. **void insert(const T& item)**
   Inserts a given item at where **cp** points at, and then it resets **cp** to point at the new item.

4. **void append(const T& item)**
   Appends a given item at the end of the list and then resets **cp** to point at the new item.

5. **void prev()**
   If **cp** is not already representing the item in the first position, it moves **cp** one step backward; otherwise, it throws a `std::logic_error`, leaving the invoking object unchanged.

6. **void next()**
   If **cp** is not already representing the item in the last position, it moves **cp** one step forward; otherwise, it throws a `std::logic_error`, leaving the invoking object unchanged.

7. **T remove()**
   Removes and returns the item **cp** points at. It resets **cp** to point at the item after the one removed, if there is one; otherwise, it resets **cp** to point at the item before the one removed, if there is one; otherwise, the the member list `line_db` has become empty, and **cp** will be reset to `db_store.end()`.

8. **void moveToFirst()**
   Resets **cp** to represent the position of the first item in the database.

9. **void moveToLast()**
   Resets **cp** to represent the position of the last item in the database.

10. **size_t getCurrentIndex() const**
    Returns the index of the item that **cp** points at. Uses 0-based indexing.

11. **void moveToIndex(size_t index)**
    Sets **cp** to represent the item at the specified index. Uses 0-based indexing.

Operations 5-11 must each throw a `std::logic_error` if the member list `db_store` is empty, leaving the invoking object unchanged.

4.2.2 Private Interface

Feel free to introduce your own private member functions in order to facilitate the operations of the other members of the class.

4.3 Resources

- list
- advance
- distance
- next
- prev

5 Class LineManager

Acting as an intermediary between the user and a database of text lines, an object of LineManager provides interactive menu-driven services to manage a number of text lines that the user enters via keyboard and/or text files.

5.1 Representation

- An object of MiniDB<std::string> representing a database of std::string items.
- An object of Menu representing a partial menu, to be displayed only when the database is or becomes empty. This menu object is set up to display the following text:

```
Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Quit
Enter an option number
??
```

- An object of Menu representing a full menu, to be displayed when the database is non-empty. This menu object is set up to display the following text:

```
Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
??
```

- Two std::size_t integers called, say, upper_span and lower_span, both initialized to the value 2; these values are used when LineManager outputs spans of lines above and below the **cp**.

5.2 Operations

5.2.1 Public Interface

1. **void run();**
   This member function launches a menu-driven session where the user can select operations that execute interactively.

   Specifically, this function must repeatedly display a menu of options until the user decides to quit. When the user decides to quit, it provides a last opportunity to save the database to a text file, if the database has been modified since the last time it was saved.

   This member function must not allow the database to be affected in any way if `MiniDB<T>` members throw `std::logic_error` exceptions. To do that, it must catch the exceptions, display the exception messages, but allow the program to continue running.

5.2.2 Private Interface

Feel free to introduce your own private member functions in order to facilitate the operations of the other members of the class.

For example, consider implementing each of the menu options in a private member function named accordingly.

5.3 Menu Options

```
1: Append input from the keyboard
```

Repeatedly prompts for and reads a text line from the keyboard and appends it at the end of the database. To end the input entry, the user must enter a single dot (.) character on a line by itself.

```
2: Insert input from the keyboard
```

Repeatedly prompts for and reads a text line from the keyboard and insert it at **cp**. To end the input entry, the user must enter a single dot (.) character on a line by itself.

```
3: Append input from text file
```

Prompts for and reads the name of a text file. It reads the lines in that file and appends them to the end of the database.

If the user file cannot be opened, it displays an appropriate error message, but it allows the program to continue running.

```
4: Insert input from text file
```

Prompts for and reads

(a) the name of a text file from the keyboard,

(b) a 1-based line index, say, n.[1]

It then reads the lines in that file and inserts them at the corresponding 0-based index $n-1$.

If n is invalid, or if the user file cannot be opened, it displays an appropriate error message, but it allows the program to continue running.

```
5: Print the current line
```

Prints the line represented by the current line.

```
6: Print a span of lines around the current position
```

Prints `upper_span` lines above the current line, followed by the line at the current line, and followed by `lower_span` lines below the current line. The last line printed will become the new current line.

```
7: Set the length of upper or lower line spans
```

Prompts for and reads an integer, say, n. If $n < 0$, it sets `upper_span` to $-n$; if $n > 0$, it sets `lower_span` to n; and if n is zero, it sets both `upper_span` and `lower_span` to zero.

If n is invalid, it displays an error message, suggests a maximum number of lines that can be included in a span of lines relative to the current line, and finally ignores the request.

```
8: Print all
```

Prints all the lines in the database, preceding each line with the index of that line. The last line printed will become the new current line.

[1] We use 1-based indexing, assuming that typical users of our `TestManager` class would find it common and intuitive to refer to the first item in a container as item 1, the second one as item 2, and so on.

The 1-based indexing interface transfers the burden of having to constantly associating item 1 with index 0, item 2 with index 1, etc. from the user to the C++/Java/etc. programmer, hence creating more work for the programmer while enhancing the friendliness of the user interface.

```
9: Move the current line
```

Prompts for and reads a 1-based integer line index, say, n, and then makes the corresponding 0-based index the current line.

If n is invalid, it displays an appropriate error message, but it allows the program to continue running.

```
10: Delete the current line
```

Removes the current line from the database. The new current line should represent

- the line that followed the line removed, if there is one; otherwise,
- the line that preceded the line removed, if there is one.

Otherwise, if the line deleted was the only line in the database, then the database would obviously become empty and the current line undefined.

```
11: Write mini-database to file
```

Prompts for and reads the name of a text file. It writes the lines in the database to the specified file.

If the user file cannot be opened, it displays an appropriate error message, but it allows the program to continue running.

```
12: Quit
```

If the database has been modified since it was last saved, it prompts the user the message "`Do you want to save changes to a file?`" and reads the user's response. If the user response is any string of characters the starts with the character `y` or `Y`, it performs as if the user selected option 11; otherwise, it terminates the program.

6 Class User

There is no need for a `User` class in C++. That's because in C++ there must always exist a `main()` free function to kick-start the execution of the program. The following `main()` function is effectively the client code in this assignment:

```
#include "LineManager.h"

int main()
{
    LineManager tm;
    tm.run();
    return 0;
}
```

7 How to organize code for a class template

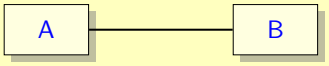
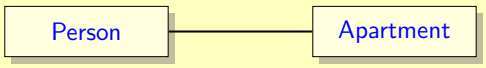
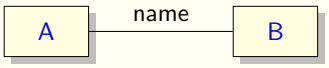
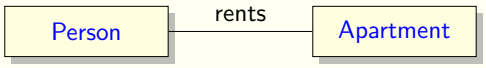
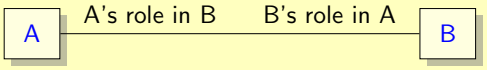
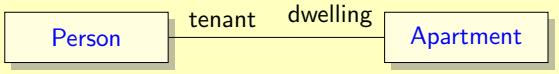
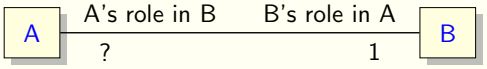
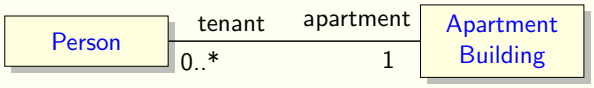
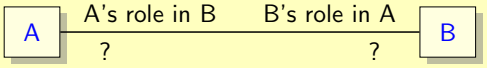
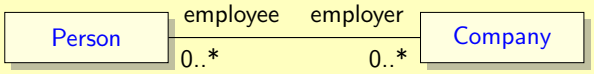
Ordinarily, you would put the class declaration in a header file, say, `Goo.h`, and put the definition of the member functions in a separate source file, say, `Goo.cpp`.

However, when you work with class templates, the ordinary rules no longer apply. That's because the compiler needs the full class template declaration and definition all in the same header file. As a result, if you define a class template in a header file `Foo.h`, be sure to include the implementation of all the member functions in that same class template in same header file; hence, NO `Foo.cpp`. Here is an example:

```
#ifndef FOO_H
#define FOO_H
#include<iostream>
using std::cout;
using std::endl;

template<typename T> //instead of T, you can use any other valid name
class Foo            // such as Type or ItemType
{
    T t1, t2;
public:
    Foo(const T&,const T&);
    void print();
};

template<typename T>
Foo<T>::Foo(const T& t1,const T& t2): t1(t1) , t2(t2){ } // T's copy ctor at work twice

template<typename T>
void Foo<T>::print()
{
    cout << "t1: " << t1 << "\nt2: " << t2 << endl; // T must have operator<<
}                                                  // overloaded
#endif /* FOO_H */
```

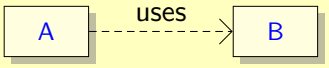
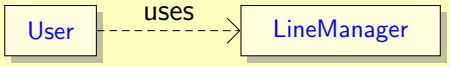
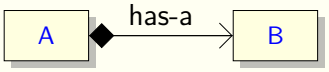
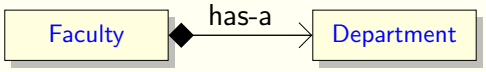
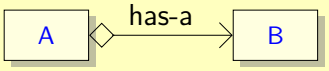
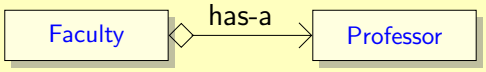
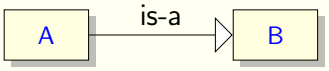
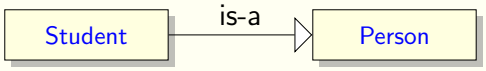
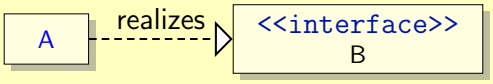
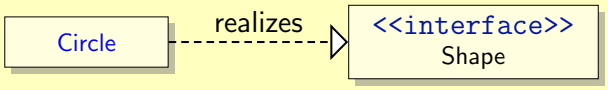
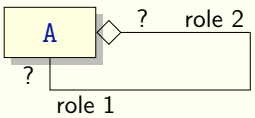
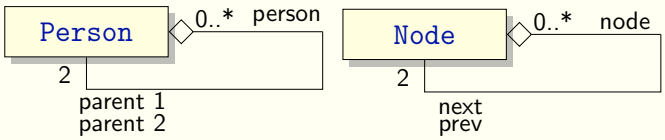
8 Summary of UML Class Diagram Notation

Bidirectional Relationship Between Objects of Classes

Association UML notation	Meaning Example
one-to-one, unnamed 	An A is associated with a B, and a B is associated with an A 
one-to-one, named 	An A is associated with a B, and a B is associated with an A 
one-to-one, named roles 	An A is associated with a B, and a B is associated with an A 
one-to-many, named roles 	An A is associated with a B, and a B is associated with many As 
many-to-many, named roles 	An A is associated with many Bs, and a B is associated with many As 

Association Multiplicity	
Notation	Association
1	only one
$0..1$	zero or one
$0..*$	zero or more
$1..*$	one or more
*	zero or more
$m..n$	many to many

Directed Relationship Between Objects of Classes

Association UML notation	Meaning Example
Dependency 	An `A` uses (depends on) a `B` but not as a data member. Basically, an `A`'s method refers to a `B`, as a local variable, a parameter, or both. 
Composition 	An `A` is made up of `B`s as data members and with lifetime dependency. The `B`s belong to `A`; if `A` is destroyed, its `B`s are destroyed as well. 
Aggregation 	An `A` is made up of `B`s as data members, but `A` does not own the `B`s. The `B`s are shared with `A`. 
Inheritance 	An `A` *is a* special kind of `B`. `A` specializes `B`. `B` generalizes `A`. `A` has access to all non-private members of `B`. 
Realization 	An `A` realizes `B`. `A` implements all the functions prototyped in B. 
Reflexive 	An association between objects of `A`. An `A` is associated with itself. 

9 Deliverables

1. Header files: `MiniDB.h`, `LineManager.h` `Menu.h`
2. Implementation files: `MiniDB.cpp`, `LineManager.cpp`, `Menu.cpp`, and `main.cpp`
3. A `README.txt` text file (as described in the course outline).

10 Marking scheme

60%	Program correctness
20%	Proper use of pointers, dynamic memory management, and C++ concepts. No C-style memory functions such as `malloc`, `alloc`, `realloc`, `free`, etc. No C-style coding.
10%	Format, clarity, completeness of output
10%	Concise documentation of nontrivial steps in code, choice of variable names, indentation and readability of program

11 Sample Program Run

```
Welcome to LineManager!
=======================

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Quit
Enter an option number
?? 2
To end input, enter a single dot and press enter
1-> notice that options 1 and 2 are equivalent
2-> when the data base is empty
3-> .

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
  2:  when the data base is empty

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
```

```
   12: Quit
Enter an option number
?? 8
 1:  notice that options 1 and 2 are equivalent
 2:  when the data base is empty

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 2:  when the data base is empty

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 10
removing the current line ...

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
```

```
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 1:  notice that options 1 and 2 are equivalent

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 10
removing the current line ...

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Quit
Enter an option number
1
To end input, enter a single dot and press enter
1-> aa
2-> bb
3-> cc
4-> dd
5-> ee
6-> .

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
```

```
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 5:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 8
 1:  aa
 2:  bb
 3:  cc
 4:  dd
 5:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
```

```
Enter an option number
?? 9
which line to move to?
3

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 3:  cc

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 2
To end input, enter a single dot and press enter
1-> xxx
2-> yyy
3-> zzz
4-> .

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
```

```
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 5:  zzz

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 8
 1:  aa
 2:  bb
 3:  xxx
 4:  yyy
 5:  zzz
 6:  cc
 7:  dd
 8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
```

```
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 9
which line to move to?
4

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
  4:  yyy

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 6
  2:  bb
  3:  xxx
  4:  yyy
  5:  zzz
  6:  cc

Choose one of the following options:
```

```
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 6:  cc

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 7
What's the length of the span
5
That's a large span of lines.  Max is 2

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
```

```
   12: Quit
Enter an option number
?? 7
What's the length of the span
2

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 6
 4:  yyy
 5:  zzz
 6:  cc
 7:  dd
 8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
```

```
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 6
  6:  cc
  7:  dd
  8:  ee
**:  EOF

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
  8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
```

```
Enter an option number
?? 7
What's the length of the span
-10
That's a large span of lines.  Max is 7

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 7
What's the length of the span
-4

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
 8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
```

```
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 6
 4:  yyy
 5:  zzz
 6:  cc
 7:  dd
 8:  ee
**:  EOF

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 9
which line to move to?
2

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
```

```
Enter an option number
?? 5
 2:  bb

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 6
**:  BOF
 1:  aa
 2:  bb
 3:  xxx
 4:  yyy

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 8
 1:  aa
 2:  bb
 3:  xxx
 4:  yyy
 5:  zzz
 6:  cc
 7:  dd
```

```
 8:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 4
Enter input filename:
C:\Users\msi\CPP\maincpp.txt

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
11:  }

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
```

```
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 8
 1:  aa
 2:  bb
 3:  xxx
 4:  yyy
 5:  zzz
 6:  cc
 7:  dd
 8:  int main()
 9:  {
10:    return 0;
11:  }
12:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 9
which line to move to?
7

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
```

```
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 4
Enter input filename:
C:\Users\msi\CPP\funcpp.txt

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 5
10:  }

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 8
 1:  aa
 2:  bb
 3:  xxx
 4:  yyy
 5:  zzz
 6:  cc
 7:  int f()
```

```
 8:  {
 9:      return 123;
10:  }
11:  dd
12:  int main()
13:  {
14:    return 0;
15:  }
16:  ee

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 11
Enter input filename:
C:\Users\msi\CPP\output.txt

Choose one of the following options:
   1: Append input from the keyboard
   2: Insert input from the keyboard
   3: Append input from text file
   4: Insert input from text file
   5: Print the current line
   6: Print a span of lines around the current position
   7: Set the length of upper or lower line spans
   8: Print all
   9: Move the current line
   10: Delete the current line
   11: Write mini-database to file
   12: Quit
Enter an option number
?? 12
goodbye
```

11.1 output.txt

```
aa
bb
xxx
yyy
zzz
cc
int f()
{
    return 123;
}
dd
int main()
{
    return 0;
}
ee
```