

1 Purpose

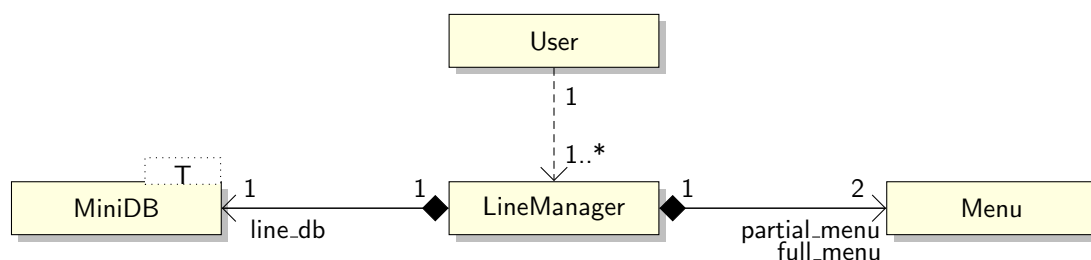
- Avoid explicit use of `new` and `delete`
- Delegate runtime storage management to the Standard Template Library (STL) containers
- Gain experience with using the STL sequential containers
- Use iterators to access and manipulate container elements
- Use code reuse to model "has-a" relationship between 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>`.

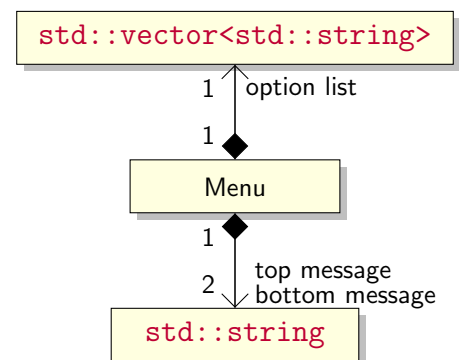
Heads up Please note that the `Menu` class in this assignment uses no `Text` objects, no arrays of `Text` objects, no doubling capacity, no `new` and `delete`, no `count` and `capacity` members, and no special member functions (copy/move constructors, copy/move assignment operator overloads, and destructor).

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.

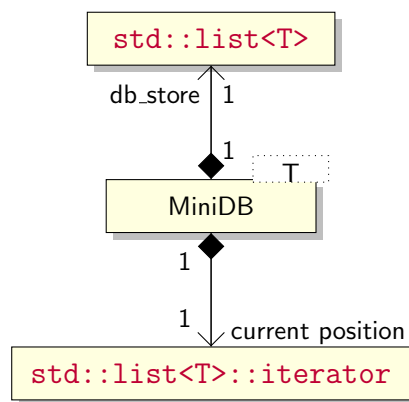
The **cp** maintained by a `MiniDB` object represents the “position” of an item that can be accessed and modified by that object. The goal of this design is to give you plenty of practice with C++ iterators, specifically, the `std::list<T>::iterator` which implements the concept of **cp** in our mini 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:

```
1 Choose one of the following options:
2   1: Append input from the keyboard
3   2: Insert input from the keyboard
4   3: Append input from text file
5   4: Insert input from text file
6   5: Quit
7 Enter an option number
8 ??
```

- 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:

```
1 Choose one of the following options:
2   1: Append input from the keyboard
3   2: Insert input from the keyboard
4   3: Append input from text file
5   4: Insert input from text file
6   5: Print the current line
7   6: Print a span of lines around the current position
8   7: Set the length of upper or lower line spans
9   8: Print all
10  9: Move the current line
11 10: Delete the current line
12 11: Write mini-database to file
13 12: Quit
14 Enter an option number
15 ??
```

- 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.

¹We 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:

```
1 #ifndef FOO_H
2 #define FOO_H
3 #include<iostream>
4 using std::cout;
5 using std::endl;
6
7 template<typename T> //instead of T, you can use any other valid name
8 class Foo           // such as Type or ItemType
9 {
10     T t1, t2;
11 public:
12     Foo(const T&,const T&);
13     void print();
14 };
15
16 template<typename T>
17 Foo<T>::Foo(const T& t1,const T& t2): t1(t1) , t2(t2){ } // T's copy ctor at work twice
18
19 template<typename T>
20 void Foo<T>::print()
21 {
22     cout << "t1: " << t1 << "\nt2: " << t2 << endl; // T must have operator<<
23 }                                                     // overloaded
24 #endif /* FOO_H */
```

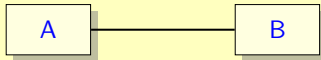
8 Summary of UML Class Diagram Notation

Bidirectional Relationship Between Objects of Classes

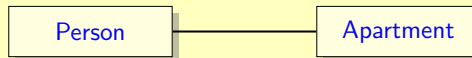
Association
UML notation

Meaning
Example

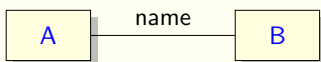
one-to-one, unnamed



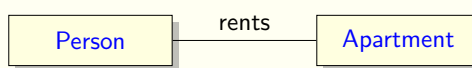
An **A** is associated with a **B**, and a **B** is associated with an **A**



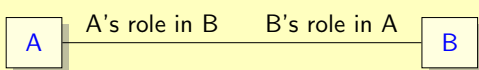
one-to-one, named



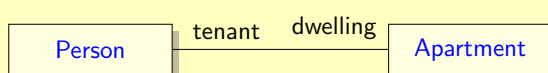
An **A** is associated with a **B**, and a **B** is associated with an **A**



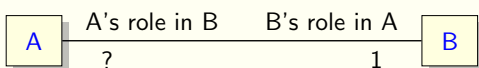
one-to-one, named roles



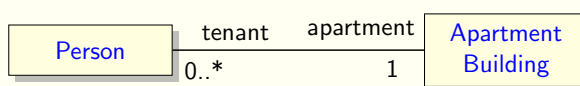
An **A** is associated with a **B**, and a **B** is associated with an **A**



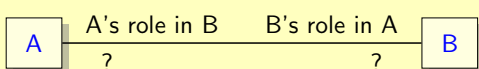
one-to-many, named roles



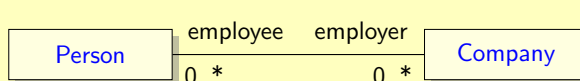
An **A** is associated with a **B**, and a **B** is associated with many **As**



many-to-many, named roles



An **A** is associated with many **Bs**, and a **B** is associated with many **As**



Association Multiplicity

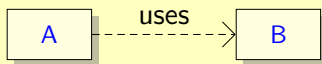
Notation	Association
1	only one
0..1	zero or one
0..*	zero or more
1..*	one or more
*	zero or more
<i>m..n</i>	many to many

Directed Relationship Between Objects of Classes

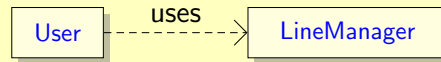
Association
UML notation

Meaning
Example

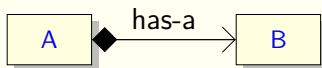
Dependency



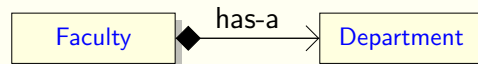
An **A** uses (depends on) a **B** but not as a data member. Basically, an **A**'s method refers to a **B**, as a local variable, a parameter, or both.



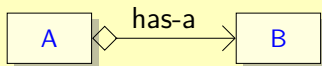
Composition



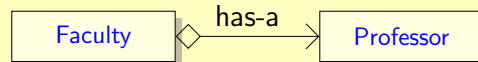
An **A** is made up of **Bs** as data members and with lifetime dependency. The **Bs** belong to **A**; if **A** is destroyed, its **Bs** are destroyed as well.



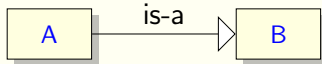
Aggregation



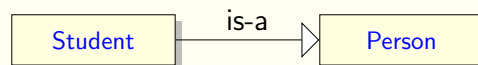
An **A** is made up of **Bs** as data members, but **A** does not own the **Bs**. The **Bs** are shared with **A**.



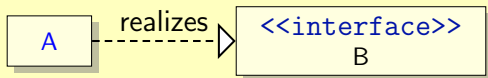
Inheritance



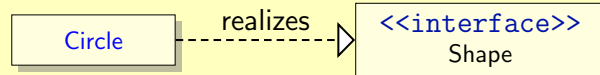
An **A** is a special kind of **B**. **A** specializes **B**. **B** generalizes **A**. **A** has access to all non-private members of **B**.



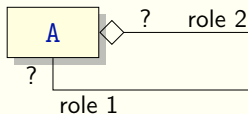
Realization



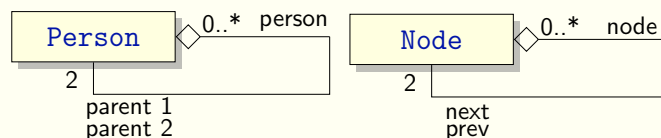
An **A** realizes **B**. **A** implements all the functions prototyped in **B**.



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 <code>malloc</code> , <code>alloc</code> , <code>realloc</code> , <code>free</code> , 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

```
1 Welcome to LineManager!
2 =====
3
4 Choose one of the following options:
5     1: Append input from the keyboard
6     2: Insert input from the keyboard
7     3: Append input from text file
8     4: Insert input from text file
9     5: Quit
10 Enter an option number
11 ?? 2
12 To end input, enter a single dot and press enter
13 1-> notice that options 1 and 2 are equivalent
14 2-> when the data base is empty
15 3-> .
16
17 Choose one of the following options:
18     1: Append input from the keyboard
19     2: Insert input from the keyboard
20     3: Append input from text file
21     4: Insert input from text file
22     5: Print the current line
23     6: Print a span of lines around the current position
24     7: Set the length of upper or lower line spans
25     8: Print all
26     9: Move the current line
27    10: Delete the current line
28    11: Write mini-database to file
29    12: Quit
30 Enter an option number
31 ?? 5
32 2: when the data base is empty
33
34 Choose one of the following options:
35     1: Append input from the keyboard
36     2: Insert input from the keyboard
37     3: Append input from text file
38     4: Insert input from text file
39     5: Print the current line
40     6: Print a span of lines around the current position
41     7: Set the length of upper or lower line spans
42     8: Print all
43     9: Move the current line
44    10: Delete the current line
45    11: Write mini-database to file
```

```

46     12: Quit
47 Enter an option number
48 ?? 8
49 1: notice that options 1 and 2 are equivalent
50 2: when the data base is empty
51
52 Choose one of the following options:
53 1: Append input from the keyboard
54 2: Insert input from the keyboard
55 3: Append input from text file
56 4: Insert input from text file
57 5: Print the current line
58 6: Print a span of lines around the current position
59 7: Set the length of upper or lower line spans
60 8: Print all
61 9: Move the current line
62 10: Delete the current line
63 11: Write mini-database to file
64 12: Quit
65 Enter an option number
66 ?? 5
67 2: when the data base is empty
68
69 Choose one of the following options:
70 1: Append input from the keyboard
71 2: Insert input from the keyboard
72 3: Append input from text file
73 4: Insert input from text file
74 5: Print the current line
75 6: Print a span of lines around the current position
76 7: Set the length of upper or lower line spans
77 8: Print all
78 9: Move the current line
79 10: Delete the current line
80 11: Write mini-database to file
81 12: Quit
82 Enter an option number
83 ?? 10
84 removing the current line ...
85
86 Choose one of the following options:
87 1: Append input from the keyboard
88 2: Insert input from the keyboard
89 3: Append input from text file
90 4: Insert input from text file
91 5: Print the current line
92 6: Print a span of lines around the current position

```



```

93 7: Set the length of upper or lower line spans
94 8: Print all
95 9: Move the current line
96 10: Delete the current line
97 11: Write mini-database to file
98 12: Quit
99 Enter an option number
100 ?? 5
101 1: notice that options 1 and 2 are equivalent
102
103 Choose one of the following options:
104 1: Append input from the keyboard
105 2: Insert input from the keyboard
106 3: Append input from text file
107 4: Insert input from text file
108 5: Print the current line
109 6: Print a span of lines around the current position
110 7: Set the length of upper or lower line spans
111 8: Print all
112 9: Move the current line
113 10: Delete the current line
114 11: Write mini-database to file
115 12: Quit
116 Enter an option number
117 ?? 10
118 removing the current line ...
119
120 Choose one of the following options:
121 1: Append input from the keyboard
122 2: Insert input from the keyboard
123 3: Append input from text file
124 4: Insert input from text file
125 5: Quit
126 Enter an option number
127 1
128 To end input, enter a single dot and press enter
129 1-> aa
130 2-> bb
131 3-> cc
132 4-> dd
133 5-> ee
134 6-> .
135
136 Choose one of the following options:
137 1: Append input from the keyboard
138 2: Insert input from the keyboard
139 3: Append input from text file

```

```

140 4: Insert input from text file
141 5: Print the current line
142 6: Print a span of lines around the current position
143 7: Set the length of upper or lower line spans
144 8: Print all
145 9: Move the current line
146 10: Delete the current line
147 11: Write mini-database to file
148 12: Quit
149 Enter an option number
150 ?? 5
151 5: ee
152
153 Choose one of the following options:
154 1: Append input from the keyboard
155 2: Insert input from the keyboard
156 3: Append input from text file
157 4: Insert input from text file
158 5: Print the current line
159 6: Print a span of lines around the current position
160 7: Set the length of upper or lower line spans
161 8: Print all
162 9: Move the current line
163 10: Delete the current line
164 11: Write mini-database to file
165 12: Quit
166 Enter an option number
167 ?? 8
168 1: aa
169 2: bb
170 3: cc
171 4: dd
172 5: ee
173
174 Choose one of the following options:
175 1: Append input from the keyboard
176 2: Insert input from the keyboard
177 3: Append input from text file
178 4: Insert input from text file
179 5: Print the current line
180 6: Print a span of lines around the current position
181 7: Set the length of upper or lower line spans
182 8: Print all
183 9: Move the current line
184 10: Delete the current line
185 11: Write mini-database to file
186 12: Quit

```

```

187 Enter an option number
188 ?? 9
189 which line to move to?
190 3
191
192 Choose one of the following options:
193     1: Append input from the keyboard
194     2: Insert input from the keyboard
195     3: Append input from text file
196     4: Insert input from text file
197     5: Print the current line
198     6: Print a span of lines around the current position
199     7: Set the length of upper or lower line spans
200     8: Print all
201     9: Move the current line
202    10: Delete the current line
203    11: Write mini-database to file
204    12: Quit
205 Enter an option number
206 ?? 5
207 3: cc
208
209 Choose one of the following options:
210     1: Append input from the keyboard
211     2: Insert input from the keyboard
212     3: Append input from text file
213     4: Insert input from text file
214     5: Print the current line
215     6: Print a span of lines around the current position
216     7: Set the length of upper or lower line spans
217     8: Print all
218     9: Move the current line
219    10: Delete the current line
220    11: Write mini-database to file
221    12: Quit
222 Enter an option number
223 ?? 2
224 To end input, enter a single dot and press enter
225 1-> xxx
226 2-> yyy
227 3-> zzz
228 4-> .
229
230 Choose one of the following options:
231     1: Append input from the keyboard
232     2: Insert input from the keyboard
233     3: Append input from text file

```

```

234     4: Insert input from text file
235     5: Print the current line
236     6: Print a span of lines around the current position
237     7: Set the length of upper or lower line spans
238     8: Print all
239     9: Move the current line
240    10: Delete the current line
241    11: Write mini-database to file
242    12: Quit
243 Enter an option number
244 ?? 5
245 5: zzz
246
247 Choose one of the following options:
248     1: Append input from the keyboard
249     2: Insert input from the keyboard
250     3: Append input from text file
251     4: Insert input from text file
252     5: Print the current line
253     6: Print a span of lines around the current position
254     7: Set the length of upper or lower line spans
255     8: Print all
256     9: Move the current line
257    10: Delete the current line
258    11: Write mini-database to file
259    12: Quit
260 Enter an option number
261 ?? 8
262 1: aa
263 2: bb
264 3: xxx
265 4: yyy
266 5: zzz
267 6: cc
268 7: dd
269 8: ee
270
271 Choose one of the following options:
272     1: Append input from the keyboard
273     2: Insert input from the keyboard
274     3: Append input from text file
275     4: Insert input from text file
276     5: Print the current line
277     6: Print a span of lines around the current position
278     7: Set the length of upper or lower line spans
279     8: Print all
280     9: Move the current line

```

```

281     10: Delete the current line
282     11: Write mini-database to file
283     12: Quit
284 Enter an option number
285 ?? 9
286 which line to move to?
287 4
288
289 Choose one of the following options:
290     1: Append input from the keyboard
291     2: Insert input from the keyboard
292     3: Append input from text file
293     4: Insert input from text file
294     5: Print the current line
295     6: Print a span of lines around the current position
296     7: Set the length of upper or lower line spans
297     8: Print all
298     9: Move the current line
299     10: Delete the current line
300     11: Write mini-database to file
301     12: Quit
302 Enter an option number
303 ?? 5
304 4: yyy
305
306 Choose one of the following options:
307     1: Append input from the keyboard
308     2: Insert input from the keyboard
309     3: Append input from text file
310     4: Insert input from text file
311     5: Print the current line
312     6: Print a span of lines around the current position
313     7: Set the length of upper or lower line spans
314     8: Print all
315     9: Move the current line
316     10: Delete the current line
317     11: Write mini-database to file
318     12: Quit
319 Enter an option number
320 ?? 6
321 2: bb
322 3: xxx
323 4: yyy
324 5: zzz
325 6: cc
326
327 Choose one of the following options:

```

```

328 1: Append input from the keyboard
329 2: Insert input from the keyboard
330 3: Append input from text file
331 4: Insert input from text file
332 5: Print the current line
333 6: Print a span of lines around the current position
334 7: Set the length of upper or lower line spans
335 8: Print all
336 9: Move the current line
337 10: Delete the current line
338 11: Write mini-database to file
339 12: Quit
340 Enter an option number
341 ?? 5
342 6: cc
343
344 Choose one of the following options:
345 1: Append input from the keyboard
346 2: Insert input from the keyboard
347 3: Append input from text file
348 4: Insert input from text file
349 5: Print the current line
350 6: Print a span of lines around the current position
351 7: Set the length of upper or lower line spans
352 8: Print all
353 9: Move the current line
354 10: Delete the current line
355 11: Write mini-database to file
356 12: Quit
357 Enter an option number
358 ?? 7
359 What's the length of the span
360 5
361 That's a large span of lines. Max is 2
362
363 Choose one of the following options:
364 1: Append input from the keyboard
365 2: Insert input from the keyboard
366 3: Append input from text file
367 4: Insert input from text file
368 5: Print the current line
369 6: Print a span of lines around the current position
370 7: Set the length of upper or lower line spans
371 8: Print all
372 9: Move the current line
373 10: Delete the current line
374 11: Write mini-database to file

```

```

375     12: Quit
376 Enter an option number
377 ?? 7
378 What's the length of the span
379 2
380
381 Choose one of the following options:
382     1: Append input from the keyboard
383     2: Insert input from the keyboard
384     3: Append input from text file
385     4: Insert input from text file
386     5: Print the current line
387     6: Print a span of lines around the current position
388     7: Set the length of upper or lower line spans
389     8: Print all
390     9: Move the current line
391    10: Delete the current line
392    11: Write mini-database to file
393    12: Quit
394 Enter an option number
395 ?? 6
396 4: yyy
397 5: zzz
398 6: cc
399 7: dd
400 8: ee
401
402 Choose one of the following options:
403     1: Append input from the keyboard
404     2: Insert input from the keyboard
405     3: Append input from text file
406     4: Insert input from text file
407     5: Print the current line
408     6: Print a span of lines around the current position
409     7: Set the length of upper or lower line spans
410     8: Print all
411     9: Move the current line
412    10: Delete the current line
413    11: Write mini-database to file
414    12: Quit
415 Enter an option number
416 ?? 5
417 8: ee
418
419 Choose one of the following options:
420     1: Append input from the keyboard
421     2: Insert input from the keyboard

```

```

422 3: Append input from text file
423 4: Insert input from text file
424 5: Print the current line
425 6: Print a span of lines around the current position
426 7: Set the length of upper or lower line spans
427 8: Print all
428 9: Move the current line
429 10: Delete the current line
430 11: Write mini-database to file
431 12: Quit
432 Enter an option number
433 ?? 6
434 6: cc
435 7: dd
436 8: ee
437 **: EOF
438
439 Choose one of the following options:
440 1: Append input from the keyboard
441 2: Insert input from the keyboard
442 3: Append input from text file
443 4: Insert input from text file
444 5: Print the current line
445 6: Print a span of lines around the current position
446 7: Set the length of upper or lower line spans
447 8: Print all
448 9: Move the current line
449 10: Delete the current line
450 11: Write mini-database to file
451 12: Quit
452 Enter an option number
453 ?? 5
454 8: ee
455
456 Choose one of the following options:
457 1: Append input from the keyboard
458 2: Insert input from the keyboard
459 3: Append input from text file
460 4: Insert input from text file
461 5: Print the current line
462 6: Print a span of lines around the current position
463 7: Set the length of upper or lower line spans
464 8: Print all
465 9: Move the current line
466 10: Delete the current line
467 11: Write mini-database to file
468 12: Quit

```



```

469 Enter an option number
470 ?? 7
471 What's the length of the span
472 -10
473 That's a large span of lines. Max is 7
474
475 Choose one of the following options:
476     1: Append input from the keyboard
477     2: Insert input from the keyboard
478     3: Append input from text file
479     4: Insert input from text file
480     5: Print the current line
481     6: Print a span of lines around the current position
482     7: Set the length of upper or lower line spans
483     8: Print all
484     9: Move the current line
485    10: Delete the current line
486    11: Write mini-database to file
487    12: Quit
488 Enter an option number
489 ?? 7
490 What's the length of the span
491 -4
492
493 Choose one of the following options:
494     1: Append input from the keyboard
495     2: Insert input from the keyboard
496     3: Append input from text file
497     4: Insert input from text file
498     5: Print the current line
499     6: Print a span of lines around the current position
500     7: Set the length of upper or lower line spans
501     8: Print all
502     9: Move the current line
503    10: Delete the current line
504    11: Write mini-database to file
505    12: Quit
506 Enter an option number
507 ?? 5
508 8: ee
509
510 Choose one of the following options:
511     1: Append input from the keyboard
512     2: Insert input from the keyboard
513     3: Append input from text file
514     4: Insert input from text file
515     5: Print the current line

```

```

516     6: Print a span of lines around the current position
517     7: Set the length of upper or lower line spans
518     8: Print all
519     9: Move the current line
520    10: Delete the current line
521    11: Write mini-database to file
522    12: Quit
523 Enter an option number
524 ?? 6
525 4: yyy
526 5: zzz
527 6: cc
528 7: dd
529 8: ee
530 **: EOF
531
532 Choose one of the following options:
533     1: Append input from the keyboard
534     2: Insert input from the keyboard
535     3: Append input from text file
536     4: Insert input from text file
537     5: Print the current line
538     6: Print a span of lines around the current position
539     7: Set the length of upper or lower line spans
540     8: Print all
541     9: Move the current line
542    10: Delete the current line
543    11: Write mini-database to file
544    12: Quit
545 Enter an option number
546 ?? 9
547 which line to move to?
548 2
549
550 Choose one of the following options:
551     1: Append input from the keyboard
552     2: Insert input from the keyboard
553     3: Append input from text file
554     4: Insert input from text file
555     5: Print the current line
556     6: Print a span of lines around the current position
557     7: Set the length of upper or lower line spans
558     8: Print all
559     9: Move the current line
560    10: Delete the current line
561    11: Write mini-database to file
562    12: Quit

```

```

563 Enter an option number
564 ?? 5
565 2: bb
566
567 Choose one of the following options:
568 1: Append input from the keyboard
569 2: Insert input from the keyboard
570 3: Append input from text file
571 4: Insert input from text file
572 5: Print the current line
573 6: Print a span of lines around the current position
574 7: Set the length of upper or lower line spans
575 8: Print all
576 9: Move the current line
577 10: Delete the current line
578 11: Write mini-database to file
579 12: Quit
580 Enter an option number
581 ?? 6
582 **: BOF
583 1: aa
584 2: bb
585 3: xxx
586 4: yyy
587
588 Choose one of the following options:
589 1: Append input from the keyboard
590 2: Insert input from the keyboard
591 3: Append input from text file
592 4: Insert input from text file
593 5: Print the current line
594 6: Print a span of lines around the current position
595 7: Set the length of upper or lower line spans
596 8: Print all
597 9: Move the current line
598 10: Delete the current line
599 11: Write mini-database to file
600 12: Quit
601 Enter an option number
602 ?? 8
603 1: aa
604 2: bb
605 3: xxx
606 4: yyy
607 5: zzz
608 6: cc
609 7: dd

```

```

610 8: ee
611
612 Choose one of the following options:
613     1: Append input from the keyboard
614     2: Insert input from the keyboard
615     3: Append input from text file
616     4: Insert input from text file
617     5: Print the current line
618     6: Print a span of lines around the current position
619     7: Set the length of upper or lower line spans
620     8: Print all
621     9: Move the current line
622    10: Delete the current line
623    11: Write mini-database to file
624    12: Quit
625 Enter an option number
626 ?? 4
627 Enter input filename:
628 C:\Users\msi\CPP\maincpp.txt
629
630 Choose one of the following options:
631     1: Append input from the keyboard
632     2: Insert input from the keyboard
633     3: Append input from text file
634     4: Insert input from text file
635     5: Print the current line
636     6: Print a span of lines around the current position
637     7: Set the length of upper or lower line spans
638     8: Print all
639     9: Move the current line
640    10: Delete the current line
641    11: Write mini-database to file
642    12: Quit
643 Enter an option number
644 ?? 5
645 11: }
646
647 Choose one of the following options:
648     1: Append input from the keyboard
649     2: Insert input from the keyboard
650     3: Append input from text file
651     4: Insert input from text file
652     5: Print the current line
653     6: Print a span of lines around the current position
654     7: Set the length of upper or lower line spans
655     8: Print all
656     9: Move the current line

```

```

657     10: Delete the current line
658     11: Write mini-database to file
659     12: Quit
660 Enter an option number
661 ?? 8
662 1: aa
663 2: bb
664 3: xxx
665 4: yyy
666 5: zzz
667 6: cc
668 7: dd
669 8: int main()
670 9: {
671 10:     return 0;
672 11: }
673 12: ee
674
675 Choose one of the following options:
676 1: Append input from the keyboard
677 2: Insert input from the keyboard
678 3: Append input from text file
679 4: Insert input from text file
680 5: Print the current line
681 6: Print a span of lines around the current position
682 7: Set the length of upper or lower line spans
683 8: Print all
684 9: Move the current line
685 10: Delete the current line
686 11: Write mini-database to file
687 12: Quit
688 Enter an option number
689 ?? 9
690 which line to move to?
691 7
692
693 Choose one of the following options:
694 1: Append input from the keyboard
695 2: Insert input from the keyboard
696 3: Append input from text file
697 4: Insert input from text file
698 5: Print the current line
699 6: Print a span of lines around the current position
700 7: Set the length of upper or lower line spans
701 8: Print all
702 9: Move the current line
703 10: Delete the current line

```

```

704     11: Write mini-database to file
705     12: Quit
706 Enter an option number
707 ?? 4
708 Enter input filename:
709 C:\Users\msi\CPP\funcpp.txt
710
711 Choose one of the following options:
712     1: Append input from the keyboard
713     2: Insert input from the keyboard
714     3: Append input from text file
715     4: Insert input from text file
716     5: Print the current line
717     6: Print a span of lines around the current position
718     7: Set the length of upper or lower line spans
719     8: Print all
720     9: Move the current line
721    10: Delete the current line
722    11: Write mini-database to file
723    12: Quit
724 Enter an option number
725 ?? 5
726 10: }
727
728 Choose one of the following options:
729     1: Append input from the keyboard
730     2: Insert input from the keyboard
731     3: Append input from text file
732     4: Insert input from text file
733     5: Print the current line
734     6: Print a span of lines around the current position
735     7: Set the length of upper or lower line spans
736     8: Print all
737     9: Move the current line
738    10: Delete the current line
739    11: Write mini-database to file
740    12: Quit
741 Enter an option number
742 ?? 8
743 1: aa
744 2: bb
745 3: xxx
746 4: yyy
747 5: zzz
748 6: cc
749 7: int f()

```

```

750 8: {
751 9:     return 123;
752 10: }
753 11: dd
754 12: int main()
755 13: {
756 14:     return 0;
757 15: }
758 16: ee
759
760 Choose one of the following options:
761 1: Append input from the keyboard
762 2: Insert input from the keyboard
763 3: Append input from text file
764 4: Insert input from text file
765 5: Print the current line
766 6: Print a span of lines around the current position
767 7: Set the length of upper or lower line spans
768 8: Print all
769 9: Move the current line
770 10: Delete the current line
771 11: Write mini-database to file
772 12: Quit
773 Enter an option number
774 ?? 11
775 Enter input filename:
776 C:\Users\msi\CPP\output.txt
777
778 Choose one of the following options:
779 1: Append input from the keyboard
780 2: Insert input from the keyboard
781 3: Append input from text file
782 4: Insert input from text file
783 5: Print the current line
784 6: Print a span of lines around the current position
785 7: Set the length of upper or lower line spans
786 8: Print all
787 9: Move the current line
788 10: Delete the current line
789 11: Write mini-database to file
790 12: Quit
791 Enter an option number
792 ?? 12
793 goodbye

```

11.1 output.txt

```
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
```