

Lab2: MiniEd

Ed是一个非常古老的行文本编辑器，曾经被用于文本文件的创建、显示、更改和其他一些操作。

Ed有两种模式，刚运行时Ed处于命令模式（command mode）。此时输入到Ed的文本会被当做命令进行处理。通过一些特殊命令，Ed可以被切换到输入模式（input mode），在输入模式下输入的文本会被当做文件内容进行保存。

本次Lab的任务是使用C++通过面向对象的方法完成一个简化版的Ed编辑器MiniEd。在MiniEd中，仅需实现Ed命令的一个子集。

命令与符号解释

在Ed命令中每个`?`为一个数字`n`，表示第`n`行，其值应该在第一行到最后一行之间（包含第一行和最后一行）。

每个`?,?`表示一个区间，其中`?`的含义同上。如`1,2`表示第一行到第二行；`5,11`表示第五行到第十一行；`6,6`表示第六行。应保证区间起点不大于终点。

每个命令为一行输入，以下为需要实现的命令说明。

- `a`

Appends text to the buffer after the current address line. If the current address is zero, the entered text is placed at the beginning of the buffer. Text is entered in input mode. The current address is set to the address of the last line entered or, if there were none, the current address is not changed.

- `?,?d`

Deletes the addressed lines from the buffer. The current address is set to the new address of the line after the last line deleted; if the lines deleted were originally at the end of the buffer, the current address is set to the address of the new last line; if no lines remain in the buffer, the current address is set to zero.

- `i`

Inserts text in the buffer before the current addressed line. If the current address is zero, the entered text is placed at the beginning of the buffer. Text is entered in input mode. The current address is set to the address of the last line entered or, if there were none, the current address is not changed.

- `?,?n`

Number command. Prints the addressed lines, preceding each line by its line number and a `<tab>` (`'\t'`). The current address is set to the address of the last line printed. Specially, `1,$n` prints all lines in the buffer.

- `Q`

Quits ed unconditionally. Unwritten changes are discarded.

- `w file`

Writes all lines to file and prints the number of bytes written to the file. Print an error if no filename is specified. The current address is unchanged.

- `?`

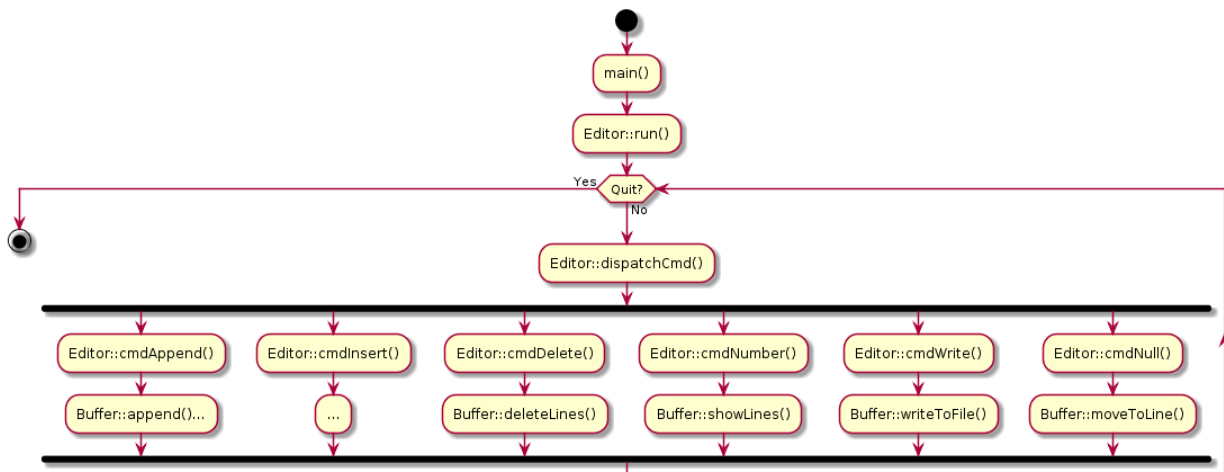
Null command. An address alone prints the addressed line. The current address is set to the address of the printed line.

对于无效指令或错误指令，请输出一个以问号开头的错误提示信息。如

```
1 | ? Bad/Unknown command
```

程序大体逻辑

MiniEd的部分代码已经给出，共五个文件：`Buffer.cc`、`Buffer.h`、`Editor.cc`、`Editor.h`和`ed.cc`。其中`Buffer.cc/h`、`Editor.cc/h`分别声明和实现了两个类。`ed.cc`中包含了`main`函数，并在其中建造并运行了一个`Editor`实例。在`Editor::run`中，程序会不断读入命令，对命令进行解析并调用相应的函数（`Editor::cmdXXX`）。`Buffer`类用于保存所有输入的文本，其中每行文本保存在一个`string`类型的字符串中。`Buffer`类有各种用于操作和展示文本的函数，供`Editor`调用。



任务

1. 实现 `Buffer::writeToFile`，将当前所有的文本保存到指定文件中，并打印所保存的字节数。
2. 实现 `Buffer::showLines`，将相应的文本行以规定格式打印出来。
3. 实现 `Buffer::insertLine`，将指定文本插入到当前行之前。
4. 实现 `Editor::cmdAppend`，完成对 `a` 命令的处理。
5. 在 `Editor::dispatchCmd` 中，完成对 `cmdWrite` 的调用。
6. 在 `Editor::dispatchCmd` 中，完成对 `1,$n` 情况的判断，以及后续的处理。（可随意更改代码，

只要实现了要求的功能即可。)

7. 完成 `Editor` 的析构函数。

回答问题：

1. 请解释在 `Editor::dispatchCmd` 中下述代码的逻辑，尤其注意对 `stringstream` 的使用。

```
1      int start, end;
2      char comma, type = ' ';
3      stringstream ss(cmd);
4      ss >> start;
5      if (ss.eof()) {
6          cmdNull(start);
7          return;
8      }
9      ss >> comma >> end >> type;
10     if (ss.good()) {
11         if (type == 'n') {
12             cmdNumber(start, end);
13             return;
14         } else if (type == 'd') {
15             cmdDelete(start, end);
16             return;
17         }
18     }
```

2. 请解释为何在 `cmdInsert` 中需要对第一行输入进行特殊处理。

编译和运行

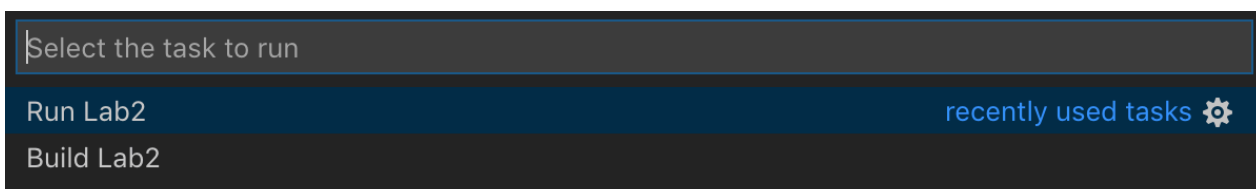
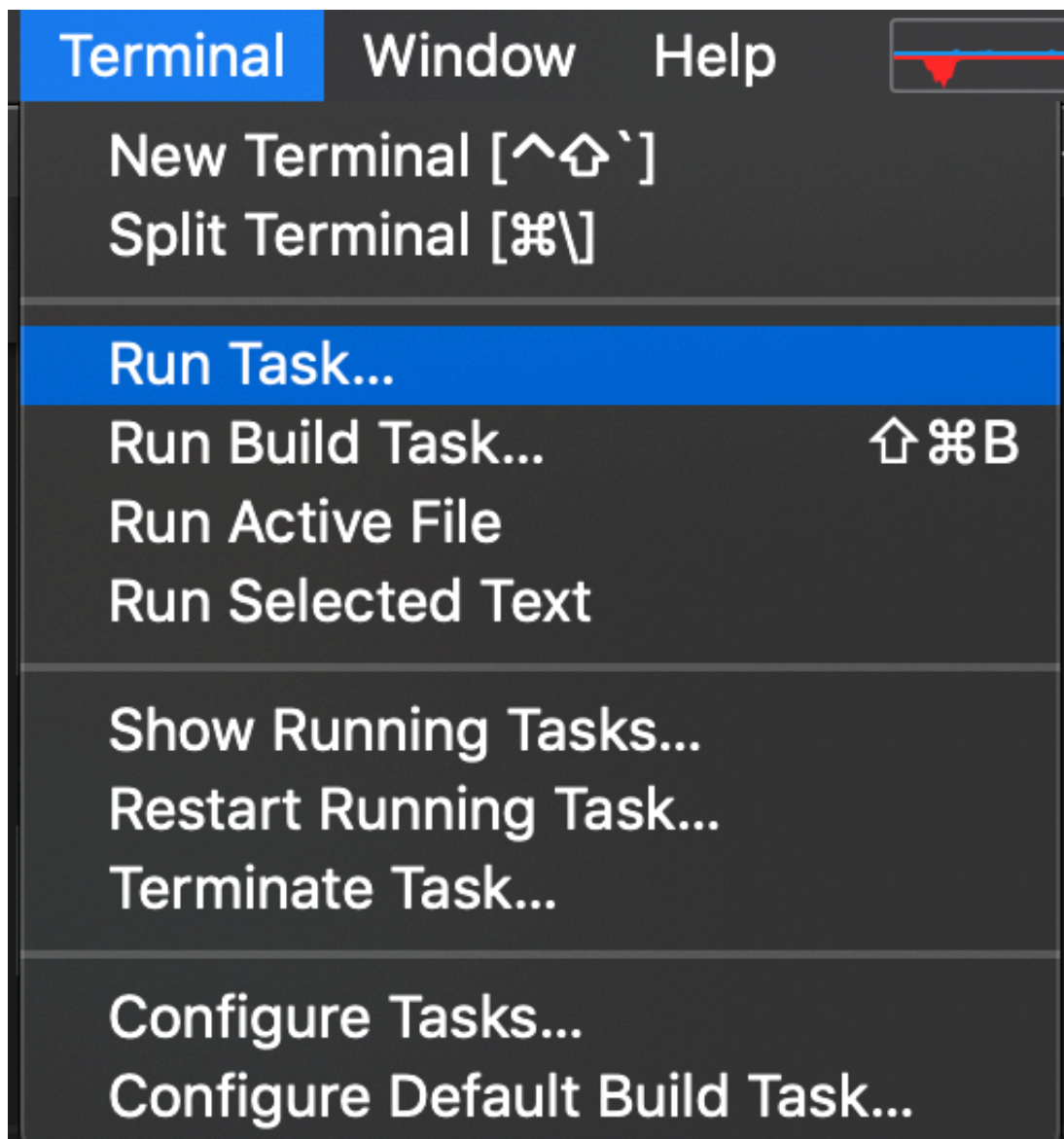
使用vscode打开lab2文件夹。

方法1

编译：在终端中，输入 `g++ -o ed Editor.cc Buffer.cc ed.cc` 并回车； 运行：在终端编译之后，输入 `./ed` 并回车。

方法2

在菜单栏中选择终端->运行任务。随后选择构建（Build Lab2）或者运行（Run Lab2）。



使用样例

```
1 cmd> a
2 It's input mode now. Quit with a line with a single dot(.)
3 #include <iostream>
4
5 int main()
6 {
7 }
8 .
9 cmd> 1,$n
10 1      #include <iostream>
11 2
```

```
12 3      int main()
13 4      {
14 5      }
15 cmd> 3
16 int main()
17 cmd> i
18 It's input mode now. Quit with a line with a single dot(.)
19 using namespace std;
20
21 #includ
22 .
23 cmd> 1,$n
24 1      #include <iostream>
25 2
26 3      using namespace std;
27 4
28 5      #includ
29 6      int main()
30 7      {
31 8      }
32 cmd> 4,5d
33 cmd> 1,$n
34 1      #include <iostream>
35 2
36 3      using namespace std;
37 4      int main()
38 5      {
39 6      }
40 cmd> 1,3c
41 ? Bad/Unknown command
42 cmd> 1,4n
43 1      #include <iostream>
44 2
45 3      using namespace std;
46 4      int main()
47 cmd> a
48 It's input mode now. Quit with a line with a single dot(.)
49     cout << "Hello my editor" << endl;
50
51     return 0;
52 .
53 cmd> 1,$n
54 1      #include <iostream>
55 2
56 3      using namespace std;
57 4      int main()
58 5      {
59 6          cout << "Hello my editor" << endl;
60 7
```

```
61 8          return 0;
62 9      }
63 cmd> write
64 ? Bad/Unknown command
65 cmd> 1,999n
66 ? Number range error
67 cmd> 4,2d
68 ? Delete range error
69 cmd> 2019
70 ? Line number out of range
71 cmd> w hello.cc
72 cmd> Q
```

提交

提交时，请将

- 你完成的MiniEd源代码压缩成zip压缩包，并重命名为 lab2-XXX.zip；
- 你对上述问题的回答 lab2-XXX.pdf。

上传到 <ftp://dmkaplony:public@public.sjtu.edu.cn:/upload/c++2019/lab2/> 中。

(其中XXX为学号，如 lab2-518037910001.zip 和 lab2-518037910001.pdf)

如果需要更改，请在文件名后加版本号，最终以最高版本号为准。如第二次提交可用 lab2-518037910001-2.zip。