Cabletime

Cabletime Linux C++ assignment

The assignment is to create a line-oriented editor like **ed** or **edlin**. Don't panic! We're not expecting you to write **emacs** in an afternoon, this is much simpler.

There is no interactive editing in this editor. Simple commands are entered to operate on a line or lines.

It is unlikely that you will be able to implement all of the listed features in the time allowed. Remember that Cabletime's interest is in seeing you produce good quality code, not churning out a whole set of functionality that sort of works.

The only command line functionality to be implemented is:
cted [filename] allowing a user to specify the file to be edited.

The commands that the editor should understand are:

- o – open a file.
- s – save the file.
  s *name* – save the file with a new name.
- q – exit the editor.
- l – list all of the lines.
  l *start* – list all of the lines from *start* onwards.
  l *start end* – list the lines from *start* to *end.*
  Note: There is no need to paginate the listing. If the user asks for too many lines then the output will scroll and the earlier lines will not be visible.
- a – append a line to the end of the current lines. Followed by a prompt for the new line.
- i *line* – insert a line at the given position. Followed by a prompt for the new line.
- d *line* – delete the line at the given position.
- c *line* – change line for the new line given by the user. Followed by a prompt for the new line.
- m *from to* – move a line from the *from* line to the *to* line.
- w – count the number of words in the current lines.
- f *string* – find the given string in the current lines.
- r *line find replace* – on the given line replace the *find* string with the *replace* string.
- r *find replace* – in all of the lines replace the *find* string with the *replace* string.

Hint:
This assignment has been given as a C test. In C++ things like the STL collections and strings make the job much easier.

Calletine

~~Wiley~~ interview questions

C questions
----

1. What does the following function do?

```
        int foo (char *s)
        {
                int n = 0;

                for (int i = 0 ; s [i] != 0 ; i++)
                {
                        if (s [i] >= '0' && s [i] <= '9')
                        {
                                n = n * 10 + (s [i] - '0');
                        }
                }

                return n;
        }
```

2.   What does the following program do?

```
        #include <stdio.h>

        int main ()
        {
                for (int i = 0 ; i < 5 ; i++)
                {
                        for (int j = 1 ; j < 100 ; j++)
                        {
                                printf ("%d", j);

                                if (j == 5)
                                        break;
                        }
                }

                return 0;
        }
```

3. What do each of the following do?

```
        *p++;

        (*p)++;
```

4. What is wrong with the following?

```
        float foo ()
        {
                short *p;
                float q, r;

                q = 1.23;
                p = &q;
                r = *p;
                return r;
        }
```

5. Write a declaration for an array of pointers to functions returning ints.

C++ questions
----

1. What is the difference between 'private' and 'protected' access?

2. What is wrong with the following code?

```
        class foo
        {
                private:
                        int i;

                public:
                        static int j;
        }

        foo::foo ()
        {
                i = 10;
                if (!j)
                        j = 20;
        }
```

3. What access do i, j, k, and l have in each of the sub-classes below?

```
        class foo
        {
                private:
                        int i;
                protected:
                        int j;
                public:
                        int k;
        }

        class foo1: private foo
        {
                private:
                        int l;
        }

        class foo2: public foo
        {
                protected:
                        int l;
        }
```

4. The function 'baz' in the following code may throw an exception of type EODException. Modify the code to catch this exception.

```
                foo (baz (23), 46);
```

5. What is wrong with the following?

```
        void foo (vector<baz>& v)
        {
                for (vector<baz>::const_iterator i = v.begin () ; i != v.end ()
; i++)
                {
                        if (toBeDeleted (*i))
                                v.erase (i);
                }
        }
```