LAB 1 - TASK 1 through TASK 3
vi / Constructs / Age

John Dempsey
COMP-232: Programming Languages
California State University, Channel Islands
August 25, 2025
Hard Due Date: September 5, 2025

For our first lab, we will write several short C programs. During Lab 3, I'll explain how to turn in your programs.

TASK 1. INVENTION

To edit files on a Unix system, you must master the vi editor.

In this step, you will create four text files: 1.txt, 2.txt, 3.txt, and 4.txt. Each of these files will build on the previous file.

1.txt

Using vi, create the 1.txt file to read as follows:

I've done it, I've done it!
I'm now taking COMP-232!

Hints:

To create the 1.txt file, type:

```
% vi 1.txt
```

To add text, type **i** to switch into insert mode and type in the above text.

Press **ESC** to exit insert mode.

If you make a mistake, you can move the cursor by:

Pressing the **L** key or space key to move to the right.

Pressing the **H** key to move to the left.

Pressing the **J** key to move down.

Pressing the **K** key to move up.

To change a word, type **cw** at the beginning of the word you want to correct. Type in your change and press **ESC** to exit.

To delete one character, move the cursor over the character to remove and press the **X** key.

To save your file, type **ZZ** or **:wq** to write and quit.

2.txt

Now copy 1.txt to 2.txt by typing at the Unix command prompt**:**

```
% cp 1.txt 2.txt

% ls                          ← Should see two files listed: 1.txt and 2.txt
```

Using vi, update 2.txt file to read as follows:

> **INVENTION**
>
> **I've done it, I've done it!**
> **Guess what I've done!**
> **The sun and the bulb are part of the fun.**
> **But, there is one thing wrong …**

Hints:

To add a new line above the current line, type **CAPITAL O** to open a line above. Type in your new text. To exit INSERT MODE, type **ESC** key.

To add a new line below the current line, type **O** key for open a line below the current line.

To delete a line, move the cursor over the line you want to delete and type **dd** (that's two d's in a row).

To write out your changes and quit the vi editor, you can type:

```
:wq
```

3.txt

Now copy 2.txt to 3.txt by typing:

```
% cp 2.txt 3.txt
```

Using vi, update 3.txt to read as follows:

```
Where the Sidewalk Ends

INVENTION

I've done it, I've done it!
Guess what I've done!
A light that plugs into the sun.
The sun is bright enough,
The bulb is strong enough,
There's only one thing wrong …
```

To save and quit the vi editor, you can also type:

```
ZZ
```

4.txt

Now copy 3.txt to 4.txt by typing:

```
% cp 3.txt 4.txt
```

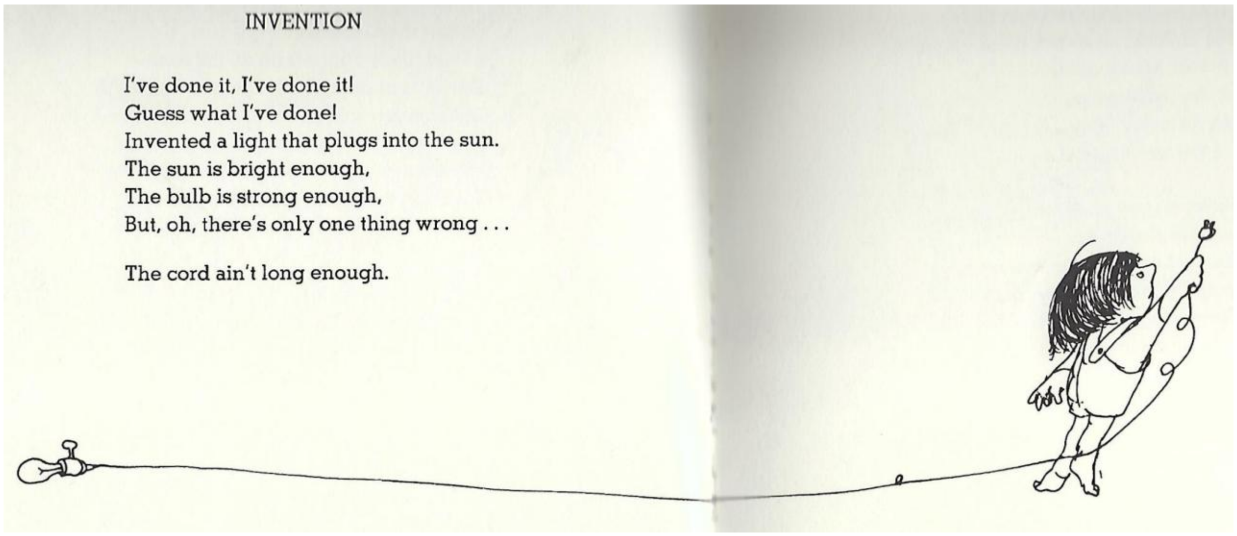
Using vi, update 4.txt to read as follows:

```
Where the Sidewalk Ends
By Shel Silverstein

INVENTION

I've done it, I've done it!
Guess what I've done!
Invented a light that plugs into the sun.
The sun is bright enough,
The bulb is strong enough,
But, oh, there's only one thing wrong …

The cord ain't long enough.
```

Save your changes.

Now type:

```
% ls            ← Should see 1.txt, 2.txt, 3.txt, and 4.txt files
% more 4.txt    ← Let's view our final version which should match
                  with the page below.
```



INVENTION

I've done it, I've done it!
Guess what I've done!
Invented a light that plugs into the sun.
The sun is bright enough,
The bulb is strong enough,
But, oh, there's only one thing wrong . . .

The cord ain't long enough.

vi Commands

To edit a file, type: % **vi filename.txt** To exit INSERT MODE, type: **ESC**

i	INSERT MODE	**l or space**	Move cursor to the right
a	Append to current position/INSERT MODE	**h**	Move cursor to the left
o	Open new line below current line/INSERT	**j**	Move cursor down
O	Capital O opens new line above/INSERT	**k**	Move cursor up
cw	Change word	**dw**	Delete word
w	Move to next word	**b**	Move back to previous word
#G	Go to line number #	**G**	Go to last line in file
H	Go to top of file.	**:w**	Write changes made. :w! overwrites
y	Yank/Copy current line into buffer	**p**	Put/Paste buffer
ZZ	Write changes and exit from vi	**:q**	Quit, exit vi. :q! Force quit
:e fn.txt	While in vi, switch file and edit fn.txt	**:e #**	Switch to previous file being edited.
gg=G	Formats your source code	**:n**	Go to next file
^Z	Push vi into background	**fg**	Bring vi back to foreground

TASK 2 Basic C Constructs

Task 2.1 myfor.c - Use a for loop to print “Hello World!” 5 times.

Using vi, edit a file called myfor.c by typing:

```
% vi myfor.c
```

Edit file myfor.c to contain:

```
int main()
{

        for (i=0; i<5; i++)
            printf(“Hello World!\n”);

}
```

To compile myfor.c, run:

```
% gcc myfor.c -o myfor
```

To run the myfor executable, type:

```
% myfor
```

TASK 2.2 mywhile.c - Use a while loop to print “Hello World!” 10 times.

Copy the myfor.c to mywhile.c using:

```
% cp myfor.c mywhile.c
```

Using vi, edit mywhile.c to print Hello World ten times using a while loop instead of a for loop.

TASK 2.3 myswitch.c - Switch/Case

Edit a file named myswitch.c.

Inside a while loop that never exits, prompt the user for a number, such as:

Enter a number between 1-4 or q for quit:

When the user enters a value, read the character using getch().

If the character entered is:

1 – print "One for the money."
2 – print "Two for the show."
3 – print "Three to get ready"
4 – print "Four for the show"

q or Q – print "Thanks for playing." ← You can use break statement to exit while loop

Anything else – print "Invalid input. Please try again."

Here's a rough start:

```
#include <stdio.h>

int main()
{
        char    ch;

        while (1) {

                printf("Enter a number between 1-4 or q to quit: ");

                ch = getchar();

                switch (ch) {

                        case '1': printf("One for the money.\n");

                        case '2': printf("Two for the show\n");

                        ….

                }

                ch = getchar();     // Read new line character
        }

}
```

Copy/paste the above code into your myswitch.c file and then type:

```
gg=G
```

to auto format your code.

You may need to change the curly quotes Word uses to regular quotes too. You may need to copy/paste the curly quotes. Type:

```
:1,$s/“/”/g
:1,$s/”/”/g
:1,$s/‘/’/g
```

The **:1,$** means “for lines 1 through the end of the file”.
s/“/”/ means switch the curly quotes to regular quotes.
And the **g** stands for global, meaning change all curly quotes in the file.

Task 2.4 myif.c – Use one large if / else statement

Copy myswitch.c to myif.c and replace the switch/case statement to use one large if / else
statement instead.

The output should be the same as for Task 2.3.

TASK 3 – How Old Are You?

This task calculates how old you are in years, months, days, hours, minutes, and seconds.

We'll assume no leap years and your age will be read in as a real number, like 19.40 years old. The .40 would represent #days/365 since your last birthday.

Here's a template which you can start with if you like:

```
john@oho:~/LAB2$ cat age.c
#include <stdio.h>

void main()
{
    float   age;

    printf("How old are you? ");
    scanf("%f", &age);

    printf("You are %.02f years old.\n", age);
}
```

Now update the above program to calculate your age in years, months, days, hours, minutes, and seconds.

```
john@oho:~/LAB2$ cat age.c

john@oho:~/LAB2$ gcc age.c -o age

john@oho:~/LAB2$ age
How old are you? 19.6
You are 19.6 years old.
You are xxx months old.
You are xxx days old.
You are xxx hours old.
You are xxx minutes old.
You are xxx seconds old.
```