**Library Module – Shared Library**

**1. Create 3 files as below.**

**· libapplication.c – will contain main() and will invoke functions in cal\_utility.c**

Created a libapplication.c file that contains main() and invoking add(), sub() functions in cal\_utility.c.

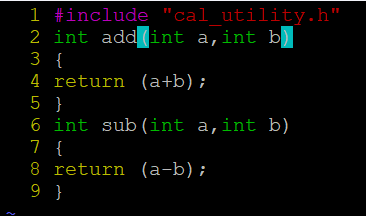
A computer screen with text on it

Description automatically generated

**· cal\_utility.c – will contain atleast 2 or more functions [ You may add definitions of the functions in this file ]**

Created a cal\_utility.c file containing definitions of add() and sub() functions.

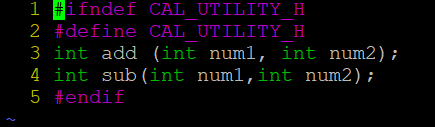
Output: The code in cal\_utility.c



**· cal\_utility.h – will contain the extern declarations/prototypes of the functions in cal\_utility.c**

Created a cal\_utility.h file containing prototypes of the functions in cal\_utility.

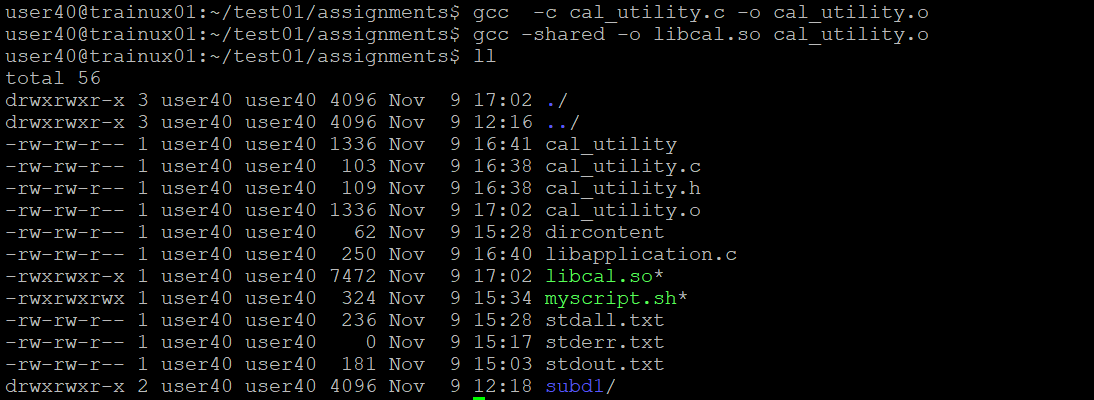
Output: The code in cal\_utility.



**2. Refer the steps for shared library and create a shared library comprising of cal\_utility.c,.h files**

To create a shared library, we need to first compile the cal\_utility.c file using -fPIC option to create a .o file. This .o file will form the shared library,

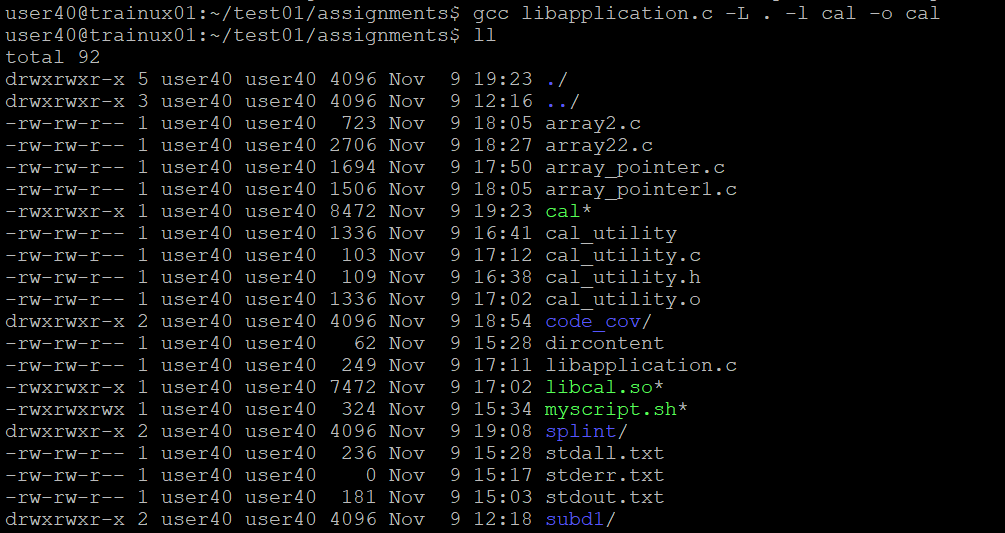
to do that use gcc command with option -shared.(Ensure that library name is prefixed with “lib” and extension as “.so”) as $ gcc -shared -o libcal.so cal\_utility.o.



**3. Create an executable using shared library.**

To create an executable file, link the library with the .o file containing the main() using command below. [Exclude the “lib” prefix and extension in library name].

$ gcc libapplication.c -L . -l cal -o cal



. Set the environment variable LD\_LIBRARY\_PATH to include library path (say if the

library path is /user/gurc4183/tools/dlib ) before running executable as below.

**Example :**

export LD\_LIBRARY\_PATH=/user40/test01/tools/dlib

**4. Execute the application created step 3.**

It is executed and the output is as fallows

Enter num1 and num2: 10 5

Addition of numbers:15

Subtraction of numbers : 5

A screen shot of a computer

Description automatically generated