Assignment 1

Vishal Bindal, 2018CS50425

Setting up ruby and python:

1) Setting up ruby

```
sudo apt-get install ruby-full
```

I’ve used the following options for my machine:

```
-I/usr/include/x86_64-linux-gnu/ruby-2.5.0 -I/usr/include/ruby-2.5.0 -lruby-2.5
```

In case there is an error like “ruby.h not found”, “ruby.config not found”, etc, then the following should be run and the above options should be replaced with the result, in the main makefile and the makefile in subdirectory 3

```
pkg-config --cflags --libs ruby-2.5
```

2) Setting up Python

```
sudo apt-get install python3
```

```
sudo apt-get install python3-dev
```

Make targets:

1) all : Creates directories obj and exe (if not present), creates .o files in subdirectory ‘obj’ for all c functions by recursively calling makefiles in all subdirectories, creates file ‘out’ in subdirectory ‘exe’.

2) run : Has exe/out as dependency, the screen is cleared, and then exe/out is run

3) clean : Removes all .o files in subdirectory ‘obj’ (if exists), and ‘out’ file in subdirectory ‘exe’ (if exists)

4) cleanAll : Removes the subdirectories obj and exe, alongwith their contents.

5) fun1 , fun2 , fun3 , fun4 : To call make only of specific functions

6) obj, exe : To create these directories

Make commands:

1) **`make`** / **`make all`**

Expected output:

```
mkdir obj

gcc -c main.c -o obj/main.o

make --directory=1

make[1]: Entering directory '/home/vishal/Desktop/cop290/Top/1'

gcc -c factorial.c -o ../obj/factorial.o

make[1]: Leaving directory '/home/vishal/Desktop/cop290/Top/1'

make --directory=2

make[1]: Entering directory '/home/vishal/Desktop/cop290/Top/2'

gcc -c randomPython.c -o ../obj/randomPython.o -I/usr/include/python3.6 -lpython3.6m

make[1]: Leaving directory '/home/vishal/Desktop/cop290/Top/2'

make --directory=3

make[1]: Entering directory '/home/vishal/Desktop/cop290/Top/3'

gcc -c rubyInC.c -o ../obj/rubyInC.o -I/usr/include/x86_64-linux-gnu/ruby-2.5.0 -I/usr/include/ruby-2.5.0 -lruby-2.5

make[1]: Leaving directory '/home/vishal/Desktop/cop290/Top/3'

make --directory=4

make[1]: Entering directory '/home/vishal/Desktop/cop290/Top/4'

gcc -c sum.c -o ../obj/sum.o

make[1]: Leaving directory '/home/vishal/Desktop/cop290/Top/4'

mkdir exe
```

```
gcc obj/rubyInC.o obj/sum.o obj/main.o obj/randomPython.o obj/factorial.o -o exe/out
-I/usr/include/python3.6 -lpython3.6m -I/usr/include/x86_64-linux-gnu/ruby-2.5.0
-I/usr/include/ruby-2.5.0 -lruby-2.5
```

2) **`make run`**

Sample output : Output of ‘make all’, followed by:

```
exe/out

Factorial of 5 is 120

This following time function is in Ruby!

Time: 2020-01-06 02:42:51 +0530

Python random number generator!

A random number between 0 and 100 is 91

Sum of 128 and 1001 is 1129
```

3) **`make clean`**, **`make cleanAll`**

4) **`make fun1`** , **`make fun1 fun2`** , etc

Functions written:-

1) factorial() in factorial.c : Returns factorial of a given natural number

2) random_python() in randomPython.c : Prints a random number in the range (0,100)

3) ruby_time() in rubyInC.c : Prints current time

4) sum() in sum.c : Returns sum of 2 integers

Makefile in subdirectories:

make / *make all* creates an object file in the directory ../obj/

make clean removes the object file from the above directory

Description of makefile in Top/:

- sub1, sub2, sub3, sub4 is for the 4 subdirectories. ‘functions’ is a variable for iterating over all of them.
- PHONY is used so that make is always called for all the subdirectories.
- Target ‘all’ has exe/out as a dependency. exe/out has the object file main.o, functions, and exe (The directory exe should exist) as dependencies
- main.o is created normally using gcc -c.
- Make is recursively called for all subdirectories using:
  - $(functions):

    ```
    $(MAKE) --directory=$@
    ```

  - Here --directory=$@ is used so that directory is changed before calling make, e.g. --directory=1, etc
- To create exe/out, the following is used:
  - gcc $(wildcard obj/*.o) -o $(outFile) -I/usr/include/python3.6 -lpython3.6m -I/usr/include/x86_64-linux-gnu/ruby-2.5.0 -I/usr/include/ruby-2.5.0 -lruby-2.5
  - Here wildcard is used for selecting all object files in the subdirectory obj. Options for python and ruby are included.
- In clean, it is checked if ‘./obj’ exists, and the files in it which match “.o” are removed. Similarly out is removed from ./exe
- In cleanAll, rm -rf is used for exe and obj
- In fun1, fun2, etc. make is called only for a specific directory. e.g. $(MAKE) --directory=1
- run has exe/out as its dependency. It clears the screen and runs exe/out

References for embedding ruby:

- https://silverhammermba.github.io/emberb/embed/
- https://stackoverflow.com/questions/626333/embedding-a-ruby-interpreter-in-a-c-app