Name：曾嘉汇

Student ID：119010415

1. How did you design your program?

1.1 In the 'main()' function, **initialize the logs**' positions by the 'rand()' function to generate a random number. Use 'logs_pos[i]' array to store the log's position in ith row.

1.2 In the 'main()' function,, use 'pthread_create()' function to **create pthreads** for wood move and frog control.

1.3 In the 'main()' function, **display the output** for the user according to the recorded variable 'status'.

1.4 In the 'logs_move()' function, use variable **'status' to control** the game progress. 'status=0' represents the game continues. 'status=1' represents the user wins. 'status=2' represents the user loses. 'status=3' represents the game exits.

1.5 In the 'logs_move()' function, the way to **move the logs** is as follows. Let the character of the end of the log becomes ' '. Depending on the direction of the log movement, change 'logs_pos[i]' and let the character of the start of the log becomes '='. Since the logs move stagger, odd rows of logs move to the left and vice versa. In addition, when the frog is on the log, the method to move the frog is similar.

1.6 In the 'logs_move()' function, check **keyboard hits to update** the frog's position like 'w' or 'W' represent the frog jumps up and so on. Clear the frog's previous position before updating. Then show the frog's current position and judge whether it jumps into the river after updating.

1.7 In the 'logs_move()' function, **check the game status** by the frog's position and then change the variable 'status' value. Use 'pthread_exit()' function to exit pthreads.

1.8 In the 'logs_move()' function, **print the map** on the screen at last.

2. The environment of running your program

The configuration matches the test environment.

```
jiahui@ubuntu:~/Downloads/source$ cat /etc/issue
Ubuntu 16.04.5 LTS \n \l

jiahui@ubuntu:~/Downloads/source$ gcc --version
gcc (Ubuntu 5.5.0-12ubuntu1) 5.5.0 20171010
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE
.
```
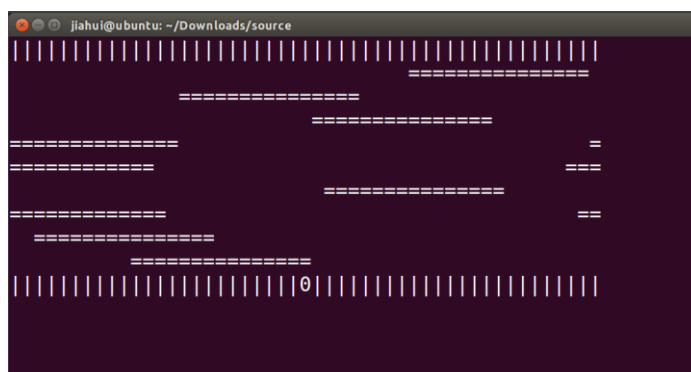
3. The steps to execute your program.

In the 'source' directory, type 'g++ hw2.cpp -lpthread' to compile the program and type

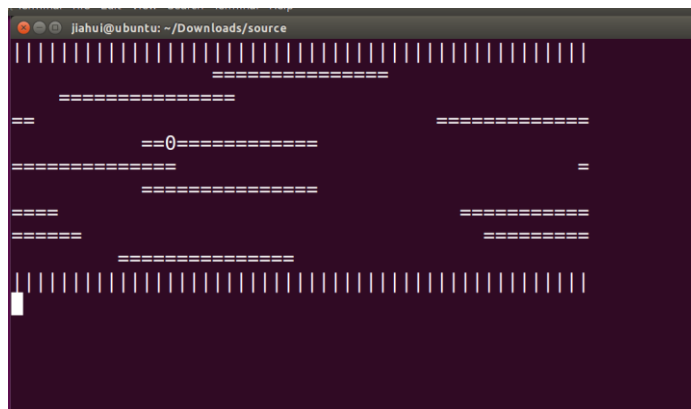'./a.out' to execute the program.

```
jiahui@ubuntu:~/Downloads/source$ g++ hw2.cpp -lpthread
jiahui@ubuntu:~/Downloads/source$ ./a.out
```

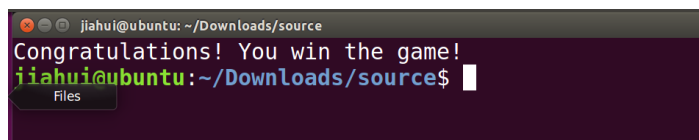## 4. Screenshot of your program output.

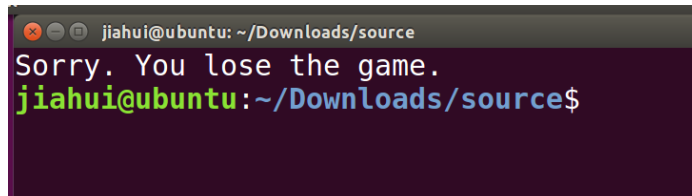The frog stands in the middle of the bottom bank of the river when the game starts.

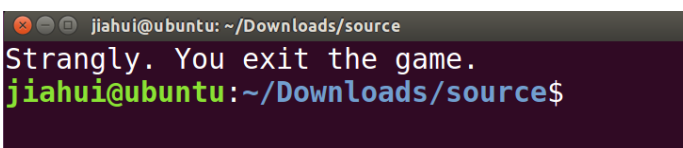The frog will move with the log when the game is in progress.

Print the message if the user wins the game.

Print the message if the user loses the game.

Print the message if the user quits the game.

5. What did you learn from the tasks?

5.1 Better understand pthread programming and how to use the API like pthread_create() and so on. Also, learn some useful functions about keyboard hit and terminal control.

5.2 Pay attention to handling the details. 'rand()' function returns a pseudo-random integral number. Thus, function 'srand()' should be used to generate some distinctive value.

5.3 Handling logs out of bounds, 'logs_pos[i]' should be updated to, for example, '(logs_pos[i]-1+COLUMN-1)%(COLUMN-1)'.

5.4 Logs move will change only the characters of the start and end of the logs.

5.5 Reminds that 'frog.x' represents which row the frog is in instead of the horizontal 'x' coordinate.

5.6 Use 'usleep(500000)' to balance the difficulty and latency.