

# READ ME BEFORE YOU DO

We have two versions of README files, one from the official `README`, the other from TA `README.md`.

For the concrete operations of your lab, you should read the `README` file, which offers you more details about your task and the rule how you should do.

But to make sure that you can use the tools provided fluently and correctly, you'd better pay a little attention to this file.

The only necessary change of codes you will make is in `bits.c` file, in which you will find all the specifications and points that you care about, and you need to use two tools already provided to check if your programs can really work correctly.

**First and most importantly, do not use Mac OS X or Windows System to finish the task.**

**You'd better run your programs on Unix or GNU/Linux (eg. Ubuntu 14.04/16.04, but I strongly suggest Ubuntu 16.04 because all the codes and tools have already been tested successfully). Using virtual machine (Vmare, VirtualBox) is a considerable choice.**

For [Vmare/VirtualBox installation](#), you should make a little effort yourself or finish it with classmate's help. And [Please click me if you are confused about how to install Ubuntu on Vmare/Virtualbox.](#)

**Once you finish one piece of program, whatever how small it is or how many point it represents, I suggest you to test if it can work well with the help of the official tool.**

There are two tools you have to use: `dlc` and `btest`. You can find the source file under your working directory.

**To make sure that the tools can work well for you, please tap:**

```
1 | ./dlc bits.c
```

It's worthy of pressing that if you have a copy of `bits.c`, on which you really work on (such as John-bits.c), you should change the file name following `./dlc`

If `compilation is successful` is given in the last line, you can make sure that `dlc` can work as assistance.

Then you can tap:

```
1 | make btest
```

If no error information is given, then the good news that `btest` can help you also comes. However, if `make btest` fails, don't worry. Try to install a `gcc-multilib` :

```
1 | sudo apt-get install gcc-multilib  
2 | password:
```

This is the situation I met and the way I solved the problem. I strongly suggest you search for best solutions through google aiming at your own problems. Take it easy and have fun!

Each function has to complete a specified task, which is given in the annotation of `bits.c` itself. Your should read it carefully and do everything in accordance with the requirement.

You don't need to push yourself to finish all the functions and get the total 41 points. Rank A could be given if your score is above 35 points. But you'd better change the best you could do.

Good luck, everybody! Please contact [cs\\_architecture@163.com](mailto:cs_architecture@163.com) if you meet any problems doing you lab.