Exercise 1: Read the page 46-47 of Lecture 2 ppt. Then, run the code in your own machine and understand the code.

Exercise 2: write a program to use bounded buffer. The program should define a buffer and can store characters. We input characters from command line, the program begin to print out all the input characters when the buffer is full. An example code is given below

```
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#define BUFFER_SIZE 15
#define TRUE 1
char *text = "Jian Zhang OS Code";
typedef struct
{
        char elem;
}item;

int main()
{
        item buffer[BUFFER_SIZE];
        int in = 0;
        int out = 0;

        item next_produced, next_comsumed;
        while (TRUE)
        {
                printf("Begin produce:\n");
                while (1)
                {
                        if (((in + 1) % BUFFER_SIZE) == out)
                        {
                                printf("\nFull.\n"); break;
                        }//do nothing

                        next_produced.elem = text[in]; // assign the text to next_produced!
                        buffer[in] = next_produced;
                        printf("%c", next_produced.elem); // added for more explanation
                        in = (in + 1) % BUFFER_SIZE;
                }

                printf("\nBegin comsume:\n");
                while (TRUE)
                {
                        if (in == out)
                        {
                                printf("\nComsume complete, it is empty now.\n"); break;
                        }//do nothing
                        next_comsumed = buffer[out];
                        printf("%c", next_comsumed.elem);//added for more explanation
                        out = (out + 1) % BUFFER_SIZE;
                        /*comsume the item in next consumed*/
                }
                break;
        }
}
```

Exercise 3: Write program A to use fork to separate the main program into two processes. In the child process, it needs to calculate the sum of 1 to 10000; in the parent process, it needs to calculate -10000 to -1. Both processes need to print out their results and exit normally.

Please see slide 11 in the lecture 3

Exercise 4: Build a program (exercise 3) stored in separate files and use makefile to compile these files. Demonstrate your understanding of using makefile.

Please see slides 24-26 in the lecture 3