x is not changing , it's used for counting the number

```
#include <stdio.h>
#include <stdlib.h>

int mutex = 1, full = 0, empty = 3, x = 0;

int wait(int);
int signal(int);
void producer();
void consumer();

int main() {
    int n;

    printf("\n1.PRODUCER\n2.CONSUMER\n3.EXIT\n");

    while (1) {
        printf("\nEnter your choice: ");
        scanf("%d", &n);

        switch (n) {
            case 1:
                if ((mutex == 1) && (empty != 0))
                    producer();
                else
                    printf("BUFFER IS FULL");
                break;

            case 2:
                if ((mutex == 1) && (full != 0))
                    consumer();
                else
                    printf("BUFFER IS EMPTY");
                break;

            case 3:
                exit(0);
                break;
        }
    }
}

int wait(int x) {
    return (--x);
}

int signal(int x) {
    return (++x);
}

void producer() {
    mutex = wait(mutex);
    full = signal(full);
    empty = wait(empty);
    x++;
```

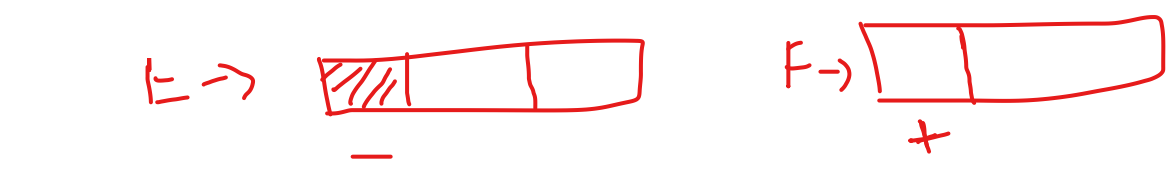
empty==0 , then it's full , empty starts from max size here 3 to 0

full=initially zero , when produces makes anything its incremented , if full=0 then it is empty

mutex decreased here from 1 to 0, no mutual exclusion allowed from here

increments value of full so that to indicate 1 item is made

since 1 item is produced therefore 1 empty space should be reduced



```
    printf("\nProducer produces the item %d", x);

    mutex = signal(mutex);
}

void consumer() {
    mutex = wait(mutex);
    full = wait(full);
    empty = signal(empty);

    printf("\nConsumer consumes item %d", x);

    x--;

    mutex = signal(mutex);
}
```

mutex is made back to 1

1 st what mutex exl have to say

2nd collect full status

3rd collect empty status , increment since 1 item is consumed

change mutex based on status