Test Cases

1. IGCreate:
   1) Normal case:
      a) `IGCreate(0)`

         (Lookup)

         Expect:

         Result:
   2) Failure case:
      a) `IGCreate(0)` //again with the same group id

         Expect: error message.

         Result: “Failed to create a group.”

      b) Use a loop to create more than 21 groups //we only support at most 20 interest groups

         Expect: error message.

         Result: "Failed to create a group at creating group 21.

2. IGPublisher:
   1) Normal case:
      a) `IGPublisher(0)`

         Expect: declare this running process as a publisher of interest group 0.

         Result: this running process is a publisher of interest group 0.

      a) `Pid=fork();` //another process, P2

         ```
         If(pid != 0){
             IGPublihser(0);
             IGPublihser(1);
         }
         ```

         Expect: declare this running child process as a publisher of interest group 0 and group 1.

         Result: this running child process is a publisher of interest group 0 and group 1.

3. IGSubscriber:
   1) Normal case:
      a) `IGPublisher(0)`

         Expect: declare this running process as a subscriber of interest group0.

         Result: this running process is a subscriber of interest group0.

      b) `Pid=fork();`

         ```
         If(pid != 0){
             IGSubscriber(0);
             IGSubscriber(1);
         }
         ```

         Expect: declare this running child process as a subscriber of interest group 0 and group 1.

         Result: this running child process is a subscriber of interest group 0 and group 1.

4. IGPublish:

1) Normal case:

a)
```
IGPublish(0, "How are you?"); //Process 1, a publisher of group 0
IGPublish(0, "How are you?"); //Process 2, a publisher of group 0
IGPublish(1, "How are you?"); //Process 2, a subscriber of group 1
```
Expect: both process 1 and 2 send messages to group 0, and process 2 sends one message to group 1.

Result: messages have been sent

2) Special case:

a)
```
for(int I = 0; I < 6; i++)
    IGPublish(0, "How are you?"); //Process 1, a publisher of group 0
```
Expect: print out "fail to publish the message"

Result: print out "fail to publish the message"

b)
```
IGPublish(0, "How are you?"); //Process 1, not a publisher of group 0
```
Expect: print out "fail to publish the message"

Result: print out "fail to publish the message"

5. IGSubscribe:

1)
```
IGSubscribe(0) //Process 1, a subscriber of group 0
IGSubscribe(0)  //Process 2, a subscriber of group 0
IGSubscribe(1)  //Process 2, a subscriber of group 1
```
Expect: p1 retrieves message from group 0, p2 retrieves message from group 0, p2 retrieves message from group 1

Result: the messages are printed out.

2) Retrieve messages when the buffer is empty

Expect: error message

Result:

3)
```
IGSubscribe(0) //Process 1, not a subscriber of group 0
```
Expect: error message

Result: