Computation structures

Problem-solving lesson 8

1. Give an implementation of the mutual exclusion between 2 processes using only the blocking message queues of size 0 as synchronization mechanism.

2. Comment the following programs. Can they be use to implement a rendezvous between two scripts?

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/msg.h>

#define MSGLEN 128
#define KEY 345782

struct {
  long mtype;
  char buf[MSGLEN];
} msg;

int main() {
  int qid;

  if ((qid=msgget(KEY,IPC_CREAT|0666))<0)
    die("couldn't access the queue");

  if (msgrcv(qid,&msg,MSGLEN,1,0)<0)
    die("failed to receive");
  printf("got '%s'\n",msg.buf);

  if (msgctl(qid,IPC_RMID,0) < 0)
    die("warning: trailing queue");

  return EXIT_SUCCESS;
}
```

```c
/** --> reuse lines 1..11 of receiver **/

int main(int argc, char ** argv) {

  if (argc != 2) {
    fprintf(stderr,
            "Usage : %s <message>\n",
            argv[0]);
    return EXIT_FAILURE;
  }

  int qid;

  if ((qid=msgget(KEY,IPC_CREAT|0666))<0)
    die("couldn't access the queue");

  msg.mtype = 1;
  strncpy(msg.buf, argv[1], MSG_LEN);
  if(msgsnd(qid,&msg,MSG_LEN,0xCAFE) < 0)
    die("failed to send");

  return EXIT_SUCCESS;
}

int die(char *msg) {
  perror(msg); exit(EXIT_FAILURE);
}
```

3. Modify the above programs in order to design a reader and a writer that communicate through a message queue:

   - The writer sends messages coming from the standard input (`stdin`) on the queue and ends by sending the " ." symbol.
   - The reader displays the messages from the queue on the standard output (`stdout`) and stops when it receives the " ." symbol .

4. Simulate a message queue using only semaphores and shared memory. For simplicity, we consider the case of only two processes sending each other integer values as messages.