۶-یک بازی کارتی متشکل از ۵ +۱ بازیکن را در نظر بگیرید. در این بازی یک کیسه وجود دارد که می‌تواند شامل حداکثر ۱۰۰ کارت باشد. روی هر کارت یک عدد صحیح بین ۱ تا ۱۰ نوشته شده است. یک بازیکن به نام boss هر بار که نوبت آن می‌شود، در صورتی که کیسه خالی باشد یک عدد کارت به آن اضافه می کند. سایر ۵ بازیکن به نام player هستند و هر موقع که نوبت یکی از آن‌ها می‌شود، در صورتی که کیسه خالی نباشد، یک کارت از کیسه خارج می‌کند و مقدار آن را به جمع مقادیر کارت هایی که تا قبل از آن از کیسه خارج کرده بود اضافه می کند. در این بازی، بازیکنی در نهایت برنده می‌شود که جمع مقادیر کارت هایی که توانسته است از کیسه خارج کند بیشتر از ۱۰۰ شود.
برنامه زیر، به صورت ناقص، برای شبیه سازی این بازی نوشته شده است:

```
int *bag, max = 100, use = 0, fill = 0, done =0;
sem_t empty, full, mutex;
void do_fill(int value) {
   bag[fill] = value;
   fill++;
   if (fill == max)
      fill = 0;}
int do_get() {
   int tmp = bag[use];
   use++;
   if (use == max)
      use = 0;
   return tmp;
}
void *boss(void *arg) {
   int i=0;
   while (1) {
      /* block A:
           i =                          ?
           sem_wait(&                   ?);
           sem_wait(&                   ?);
           if (                      ?  )
                do_fill(i);
           sem_post(&                   ?);
           sem_post(&                   ?);
      */
      if (done == 1){
         printf("Game is done (Boss)\n");
         return(NULL);}
      }
}
void *player(void *arg) {
   int sum = 0;
   while (1) {
   /* block B:
        sem_wait(&                   ?);
        sem_wait(&                   ?);
        if( done==1) printf("Game is done") else{
            sum =                                ?
   */
      if (sum > 100){
         done = 1;
         printf("I  am the winner (palyer: %lld)\n",
(long long int) arg);
       }
      /* block C:
           sem_post(&                ?);
           sem_post(&                ?);
      */
      printf("%lld %d\n", (long long int) arg, sum);
      if (done == 1){
           printf("Game is done (player: %lld)\n",(long
long int) arg);
           return(NULL);
       }//endif(done==1)
     }//endelseif(done==1)
   }//endwhile
}

int main(int argc, char *argv[]) {
   bag = (int *) malloc(max * sizeof(int));
   int i;
   /* block D:
      sem_init(&empty,          ?);
      sem_init(&full,           ?);
      sem_init(&mutex,          ?);
   */
   pthread_t bid, pid[5];
   pthread_create(&bid, NULL, boss, NULL);
   for (i = 0; i < 5; i++) {
      pthread_create(&pid[i], NULL, player, (void *)
(long long int) i);
   }
   pthread_join(bid, NULL);
   for (i = 0; i < 5; i++) {
      pthread_join(pid[i], NULL);
   }
   return 0;
}
```

الف) قسمت‌های علامت گذاری شده با ? در بخش‌های C, B, A و D را تکمیل کنید. (۵نمره)

ب) میخواهیم این بازی را توسط کرنل ماژول و برنامه‌نویسی مالتی پروسس پیاده‌سازی کنیم به صورتی که boss و playerها به جای thread توسط پروسس‌ها پیاده‌سازی شوند، کیسه کارت توسط کرنل ماژول مدیریت شود و توابع do_get و do_fill جزو APIهای کرنل ماژول باشند.
شبه کد کرنل ماژول را بنویسید. در این شبه کد، ساختار و توابع ضروری کرنل‌ماژول مشخص شود. (۵ نمره)

ج) این قسمت را بعد از امتحان نوشته و تا حداکثر سه شنبه شب ساعت ۲۳:۵۹ در یکتا آپلود کنید. (۲۰ نمره)

- کد کامل قسمت الف را نوشته از کامپایل و اجرای صحیح آن مطمین شوید.
- برای قسمت ب،کد پروسسها و کرنل ماژول را بنویسید. سپس یک اسکریپت shell بنویسید که ماژول و برنامه‌های موردنظر را کامپایل، لود و اجرا کند.