First we define ‘int state[N]’ to be used in share memory:

struct shm {

int state[N];

} \*shared\_memory;

Initialize\_shared\_memory(){

/\* to create share memory section \*/

* **shmget** … sizeof(\*shared\_memory)
* *printf(“Memory attached at shmid %d\n”, shmid);*
* **shmat** … shared\_memory = (shm\*)shmat …
* *if(…) printf(“Shmat succeed\n”);*

/\* to assign values to variable in shared memory \*/

* shared\_memory->state[i]=…

/\* to initialize semaphores \*/

* **semget** … *N+1 semaphores in a group*
* *printf(“Semaphores group id: %d\n”, sem\_group);*
* union semun -> we must define semun ourselves to use **semctl** (SETALL)

/\* to implement ‘up’ and ‘down’ functions \*/

* first define: struct sembuf *my\_sem\_b*;
* use structure fields to assign:
  + semaphore number,
  + semaphore operation (add, remove)
  + operation flags (SEM\_UNDO)
* use semop …(…, &my\_sem\_b, 1),
* if () fprintf(stderr, “semaphore failed\n”);