```
========== initsem.c =========
#include "semaphore.h"

void main(void) {
   sem_init();
}



========== semtest1.c ===========
#include "semaphore.h"
#include <stdio.h>
int main(void)
{
        int i;
        int sem[10];
        int ret;

        errno = 0;
        ret = create_sem(-4, 1);
        if (ret == -1)
          printf("Create_sem failed, errno = %d\n", errno);

        ret = create_sem(111, -3);
        if (ret == -1)
          printf("Create_sem failed, errno = %d\n", errno);

        for(i = 0; i < 10; i++) {
          errno = 0;
          sem[i] = create_sem(i, 1);
          printf("create_sem(%d, 1)
                  returns %d\terrno = %d\n", i, sem[i], errno);
        }

        for(i = 0; i < 10; i++) {
          errno = 0;
          ret = delete_sem(sem[i]);
          printf("delete_sem(%d) returns %d\n", sem[i], ret);
        }
}


==============  semtest2.c =============
#include "semaphore.h"
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <stdio.h>

#define key 111

int main(void)
{
        int ret;
        int sem;
        int i;
        int status;

        if (fork()) {
          sem = create_sem(key, 1);
          printf("Parent process has obtained semaphore %d\n", sem);
          for(i = 0; i< 3; i++) {
            sleep(2);
            printf("Parent tries to enter CS\n");
            errno = 0;
            ret = p(sem);
            printf("Parent has entered CS (ret = %d, errno = %d)\n", ret, errno)
;
            sleep(3);
            errno = 0;
            ret = v(sem);
            printf("Parent has left CS (ret = %d, errno = %d)\n", ret, errno);
          }
```



```
          wait(&status);
          errno = 0;
          ret = delete_sem(sem);
          printf("Parent has deleted sem (ret = %d, errno = %d)\n", ret, errno);
        } else {
          sem = create_sem(key, 1);
          printf("Child process has obtained semaphore %d\n", sem);
          for(i = 0; i< 3; i++) {
            sleep(1);
            printf("Child tries to enter CS\n");
            errno = 0;
            ret = p(sem);
            printf("Child has entered CS (ret = %d, errno = %d)\n", ret, errno);

            sleep(4);
            errno = 0;
            ret = v(sem);
            printf("Child has left CS (ret = %d, errno = %d)\n", ret, errno);
          }
          exit(0);
        }
}

=============== semtest3.c ===============
#include "semaphore.h"
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <stdio.h>

#define key 111

int body(int id, int sem) {
        int ret;

        printf("Child(%d) tries to enter CS\n", id);
        errno = 0;
        ret = p(sem);
        printf("Child(%d) has entered CS (ret = %d, errno = %d)\n", id, ret, err
no);
        sleep(5);
        errno = 0;
        ret = v(sem);
        printf("Child(%d) has left CS (ret = %d, errno = %d)\n", id, ret, errno)
;
        exit(id);
}

int main(void)
{
        int ret;
        int sem;
        int i;
        int status;

        sem = create_sem(key, 2);

        for (i = 0; i < 10; i++) {
          if (fork() == 0)  body(i, sem);
        }

        for (i = 0; i < 10; i++) {
                wait(&status);
                printf("Child(%d) has terminated\n", (status >> 8));
        }
}
```