

1.)

			Foglalási igény		Memóriaterület-szabad terület					
					30	35	15	25	75	45
30k, 35k, 15k, 25k, 75k, 45k			39	30	30	35	15	25	75	45
			40	30	30	35	15	25	75	45
39k, 40k, 33k, 20k, 21k			33	30	30	35	15	25	75	45
			20	30	30	35	15	25	75	45
			21	30	30	35	15	25	75	45
WORST FIT			Foglalási igény		Memóriaterület-szabad terület					
					30	35	15	25	75	45
			39	30	35	15	25	39 36	45	
			40	30	35	15	25	75	40 5	
			33	30	35	15	25	33 3	40 5	
			20	30	20 15	15	25	33 3	40 5	
			21	21 9	35	15	25	33 3	40 5	
FIRST FIT			Foglalási igény		Memóriaterület-szabad terület					
					30	35	15	25	75	45
			39	30	35	15	25	36 39	45	
			40	30	35	15	25	75	5 40	
			33	30	2 33	15	25	75	45	
			20	10 20	35	15	25	75	45	
			21	30	35	15	4 21	75	45	
NEXT FIT			Foglalási igény		Memóriaterület-szabad terület					
					30	35	15	25	75	45
			39	30	35	15	25	39 36	45	
			40	30	35	15	25	75	40 5	
			33	30	33 2	15	25	75	45	
			20	30	35	15	20 5	75	45	
			21	30	35	15	25	21 15	45	
BEST FIT			Foglalási igény		Memóriaterület-szabad terület					
					30	35	15	25	75	45
			39	30	35	15	25	75	39 6	
			40	30	35	15	25	40 35	45	
			33	30	33 2	15	25	75	45	
			20	30	35	15	20 5	75	45	
			21	21 9	35	15	25	75	45	

```

#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12 // Kulcs

int semid,      // Szemafor azonosító
    nsems,      // Szemaforok száma
    semnum,     // Szemafor száma
    rtn;        // Visszatérési érték

int semflg;     // Flag
struct sembuf sembuf, *sop;
union semun;

int cmd;        // semctl()-nek szóló parancs változója

int main()
{
    int arg;

    nsems = 1;      // Egyetlen szemafor a set-ben
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("\n Az azonosító: %d\n",semid);
    printf ("Kérem a semval értéket: ");

    semnum = 0;     // 0. szemafort azonosítom

    cmd = SETVAL;    // Állítsd be a szemafor értéket
    scanf("%d", &arg);
    rtn = semctl(semid, semnum, cmd, arg); // Semid-vel azonosított set 0-ik szemafor

    printf("\nVisszatérési érték: %d\nSemval értéke: %d\n",rtn, arg);

    return 0;
}

```

2.)

```

simon28@jerry:~/oprend/second_try/gyakl1$ gcc semset.c -o semset
simon28@jerry:~/oprend/second_try/gyakl1$ ./semset

```

```

    Az azonosító: 950283
Kérem a semval értéket: ./semset

Visszatérési érték: 0
Semval értéke: 0

```

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12 // Kulcs

    int semid,nsems,rtn;
    int semflg;
    struct sembuf sembuf, *sop;
    union semun arg;
union semun {
    int val;
    struct semid_ds *buf;
    unsigned short int *array;
    };
    int cmd;

int main()
{

    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("Azonosító: %d\n",semid);

    cmd = GETVAL; // Szemafor értéket ad vissza majd az rtn-ben
    rtn = semctl(semid,0, cmd, NULL);

    printf("Semval kiolvasott értéke: %d ",rtn);
    printf("\n");

    return 0;

}

```

```

simon28@jerry:~/oprend/second_try/gyak11$ gcc semval.c -o semval
simon28@jerry:~/oprend/second_try/gyak11$ ./semval
Azonosító: 950283
Semval kiolvasott értéke: 0
simon28@jerry:~/oprend/second_try/gyak11$ █

```

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x10 // Kulcs

    int semid, nsems, rtn;
    int semflg;
    struct sembuf sembuf, *sop;
    union semun {
        int val;
    struct semid_ds *buf;
        unsigned short int *array;
    };
    int cmd;

int main()
{

    int arg;
    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("semid értéke: %d\n",semid);

    cmd = IPC_RMID;          // Ezzel szüntetjük meg
    rtn = semctl(semid, 0, cmd, arg);

    printf("Kill visszatérés: %d\n",rtn);

    return 0;

}

```

```

simon28@jerry:~/oprend/second_try/gyak11$ gcc semkill.c -o semkill
simon28@jerry:~/oprend/second_try/gyak11$ ./semkill
semid értéke: 1409037
Kill visszatérés: 0
simon28@jerry:~/oprend/second_try/gyak11$ █

```

```

#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12 // Kulcs

int semid, nsems, rtn;
unsigned nsops; // Ezzel adjuk meg, hány struktúrával foglalkozzon
int semflg;
struct sembuf sembuf, *sop;

int main()
{

    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("Azonosító: %d\n",semid);
    printf ("\n");

    nsops = 1; //Egy művelet van
    sembuf.sem_num = 0; //A 0-ik szemaforral foglalkozunk
    sembuf.sem_op = 1; // Növelés
    sembuf.sem_flg = 0666; // Flag beállítás
    sop = &sembuf; // Argumentum kérése
    rtn = semop(semid, sop, nsops); // A 0-val visszatérő semop sikeres.
    printf("Visszatérési érték: %d\n",rtn);

    return 0;
}

```

```
simon28@jerry:~/oprend/second_try/gyak11$ gcc semup.c -o semup
```

```
simon28@jerry:~/oprend/second_try/gyak11$ ./semup
```

```
Azonosító: 950283
```

```
Visszatérési érték: 0
```

```
simon28@jerry:~/oprend/second_try/gyak11$
```

2.a)

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

sem_t m;

void* thread(void* arg)
{
    //wait
    sem_wait(&m);
    printf("A pitem: %d\nVárakozás...\n", getpid()); // PID kiíratás
    sleep(3);    // Várakozás
    sem_post(&m);
}

int main()
{
    sem_init(&m, 0, 1);
    pthread_t t1,t2,t3; // 3 feladat
    pthread_create(&t1, NULL,thread,NULL);
    pthread_create(&t2, NULL,thread,NULL);
    pthread_create(&t3, NULL,thread,NULL);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    sem_destroy(&m); // Megszüntetés
    return 0;
}
```

```
simon28@jerry:~/oprend/second_try/gyak11$ cc -pthread gyak11_2.c -o gyak11_2
simon28@jerry:~/oprend/second_try/gyak11$ ./gyak11_2
A pitem: 5114
Várakozás...
A pitem: 5114
Várakozás...
A pitem: 5114
Várakozás...
simon28@jerry:~/oprend/second_try/gyak11$
```