1.)

Az összes osztály -erőforrások száma: (10, 5, 7)						
Kiinduló állapot						
	1. lépés			**2. lépés**		
	MAX. IGÉNY			**FOGLAL**		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**
P0	7	5	3	0	1	0
P1	3	2	2	2	0	0
P2	9	0	2	3	0	2
P3	2	2	2	2	1	1
P4	4	3	3	0	0	2

Kezdeti	Foglal			Még		
álapot	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

1. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [3 3 2]

A SZABAD ki tudja elégíteni a P3-et

	3	3	2
+	2	1	1

Új SZABAD = [5 4 3]

2. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [5 4 3]

A SZABAD ki tudja elégíteni a P1-et

	5	4	3
+	2	0	0

Új SZABAD = [7 4 3]

3. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [7 4 3]

A SZABAD ki tudja elégíteni a P4-et

	7	4	3
+	0	0	2

Új SZABAD = [7 4 5]

4. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [7 4 5]

A SZABAD ki tudja elégíteni a P2-et

	7	4	5
+	3	0	2

Új SZABAD = [10 4 7]

5. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [10 4 7]

A SZABAD ki tudja elégíteni a P2-et

	10	4	7
+	0	1	0

Új SZABAD = [10 5 7]

5. Lépés	Foglal			Még		
	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	4	3
P1	2	0	0	1	2	2
P2	3	0	2	6	0	0
P3	2	1	1	0	1	1
P4	0	0	2	4	3	1

SZABAD = [10 5 7]

A rendszer **Biztonságos állapotban** van!

P3 - P1 - P4 - P2 - P0

2.)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MSGKEY 654321L

struct msgbuf1 {
        long mtype;
        char mtext[512];
} sndbuf, *msgp;



int main()
{
        int msgid;
        key_t key;
        int msgflg;
        int rtn, msgsz;

        key = MSGKEY;
        msgflg = 00666 | IPC_CREAT;
        msgid = msgget( key, msgflg);
        if ( msgid == -1) {
                perror("\n The msgget system call failed!");
                exit(-1);
        }
        printf("\n Az msgid %d, %x : ", msgid,msgid);

        msgp        = &sndbuf;
        msgp->mtype = 1;
        strcpy(msgp->mtext,"Egyik uzenet");
        msgsz       = strlen(msgp->mtext) + 1;

        rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);
        printf("\n Az 1. msgsnd visszaadott %d-t", rtn);
        printf("\n A kikuldott uzenet:%s", msgp->mtext);

        strcpy(msgp->mtext,"Masik uzenet");
        msgsz       = strlen(msgp->mtext) + 1;
        rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgsz, msgflg);
        printf("\n A 2.  msgsnd visszaadott %d-t", rtn);
        printf("\n A kikuldott uzenet:%s", msgp->mtext);
        printf("\n");

        exit (0);
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc msgcreate.c -o msgcreate
simon28@jerry:~/oprend/second_try/gyak10$ ./msgcreate

 Az msgid 557056, 88000 :
 Az 1. msgsnd visszaadott 0-t
 A kikuldott uzenet:Egyik uzenet
 A 2.  msgsnd visszaadott 0-t
 A kikuldott uzenet:Masik uzenet
simon28@jerry:~/oprend/second_try/gyak10$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MSGKEY 654321L

struct msgbuf1 {
        long mtype;
        char mtext[512];
} rcvbuf, *msgp;

struct msqid_ds ds, *buf;

int main()
{
        int msgid;
        key_t key;
        int mtype, msgflg;
        int rtn, msgsz;

        key = MSGKEY;
        msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

        msgid = msgget( key, msgflg);
        if ( msgid == -1) {
                perror("\n The msgget system call failed!");
                exit(-1);
        }
        printf("\n Az msgid: %d",msgid);

        msgp = &rcvbuf;
        buf = &ds;
        msgsz = 20;
        mtype = 0;
        rtn = msgctl(msgid,IPC_STAT,buf);
        printf("\n Az uzenetek szama: %d",buf->msg_qnum);

        while (buf->msg_qnum) {

                rtn = msgrcv(msgid,(struct msgbuf *)msgp, msgsz, mtype, msgflg);
                printf("\n Az rtn: %d,  a vett uzenet:%s\n",rtn, msgp->mtext);
                rtn = msgctl(msgid,IPC_STAT,buf);
        }
        exit (0);
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc msgrcv.c -o msgrcv
simon28@jerry:~/oprend/second_try/gyak10$ ./msgrcv

 Az msgid: 557056
 Az uzenetek szama: 2
 Az rtn: 13,  a vett uzenet:Egyik uzenet

 Az rtn: 13,  a vett uzenet:Masik uzenet
simon28@jerry:~/oprend/second_try/gyak10$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MSGKEY 654321L

int main()
{
        int msgid, msgflg,  rtn;
        key_t key;
        key = MSGKEY;
        msgflg = 00666 | IPC_CREAT;
        msgid = msgget( key, msgflg);

        rtn = msgctl(msgid, IPC_RMID, NULL);
        printf ("\n Vissztert: %d \n", rtn);

        exit (0);
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc msgctl.c -o msgctl
simon28@jerry:~/oprend/second_try/gyak10$ ./msgctl

 Vissztert: -1
simon28@jerry:~/oprend/second_try/gyak10$
```

2.a)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MAX 100


struct msg_buffer {
  long msg_type;
  char msg_text[100];
} message;

int main(void) {
    key_t key;
    int msg_id;


    key = ftok("gyakl0_2_create.c", 65);
    msg_id = msgget(key, 0664 | IPC_CREAT);
    message.msg_type = 1;

    int ok= 0;



    while (ok==0) {

        printf("Kerem az uzenetet: \n");

        fgets(message.msg_text, MAX, stdin);
        if (strcmp(message.msg_text, "exit\n")==0) {
            ok = 1;
            continue;
        }
        msgsnd(msg_id, &message, sizeof(message), 0);
        printf("Uzenet elkuldve: %s", message.msg_text);
    }



  return 0;
}
```

```
simon28@jerry:~/oprend/second_try/gyakl0$ gcc gyakl0_2a.c -o gyakl0_2a
simon28@jerry:~/oprend/second_try/gyakl0$ ./gyakl0_2a
Kerem az uzenetet:
Hello World!
Uzenet elkuldve: Hello World!
Kerem az uzenetet:
exit
simon28@jerry:~/oprend/second_try/gyakl0$ nano gyakl0_2a.c
simon28@jerry:~/oprend/second_try/gyakl0$
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MAX 100

struct msg_buffer {
  long msg_type;
  char msg_text[100];
} message;

int main(void) {
    key_t key;
    int msg_id;

    key = ftok("gyak10_2.c", 65);
    msg_id = msgget(key, 0664 | IPC_CREAT);

    struct msqid_ds stat;
    //msgctl(msg_id, IPC_RMID, NULL);
    msgctl(msg_id, IPC_STAT, &stat);
    printf("Az uzenetek darabszama jelenleg: %d\n", (int)stat.msg_qnum);
    return 0;
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc gyak10_2b.c -o gyak10_2b
simon28@jerry:~/oprend/second_try/gyak10$ ./gyak10_2b
Az uzenetek darabszama jelenleg: 1
simon28@jerry:~/oprend/second try/gyak10$
```

3.)

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>

#define SHM_KEY 123456L


int main(void) {
    int shmid;
    key_t key = SHM_KEY;
    int size = 200;
    int shmflg = 0;

    shmid = shmget(key, size, shmflg);

    if (shmid < 0) {
        shmflg = 00666 | IPC_CREAT;
        shmid = shmget(key, size, shmflg);
        if (shmid < 0) {
            perror("shmget hívás hiba\n");
            return -1;
        }
    } else {
        printf("a szegmens már létrejött\n");
    }
    printf("A kellő ID: %d\n", shmid);

  return 0;
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc shmcreate.c -o shmcreate
simon28@jerry:~/oprend/second_try/gyak10$ ./shmcreate
a szegmens már létrejött
A kellő ID: 327680
simon28@jerry:~/oprend/second try/gyak10$
```

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>

#define SHM_KEY 123456L


int main(void) {
    int shmid;
    key_t key = SHM_KEY;
    int size=200;
    int shmflg = 0;
    struct shmid_ds stat;

    if (shmid < 0) {
        perror("shmget hívás hiba\n");
        return -1;
    }

    shmid = shmget(key, size, shmflg);
    shmctl(shmid, IPC_RMID, NULL);
    shmctl(shmid, IPC_STAT, &stat);
    printf("Szegmens méret: %d\n", (int)stat.shm_segsz);
    printf("Utolso muvelet: PID %d\n", stat.shm_lpid);




  return 0;
}
```

```
simon28@jerry:~/oprend/second_try/gyak10$ gcc shmctl.c -o shmctl
simon28@jerry:~/oprend/second_try/gyak10$ ./shmctl
Szegmens méret: 512
Utolso muvelet: PID 0
simon28@jerry:~/oprend/second_try/gyak10$
```