Erőforrások száma:(10,5,7) - az összes osztály

1.lépés

	MAX. IGÉNY		
	R1	R2	R3
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3

2.lépés

	MAX. IGÉNY		
	R1	R2	R3
P0	0	1	0
P1	2	0	0
P2	3	0	2
P3	2	1	1
P4	0	0	2
	7	2	5

3.lépés

	MAX. IGÉNY		
	R1	R2	R3
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

R1 = 10 - 7 = 3

R2 = 5 - 2 = 3

R3 = 7 - 5 = 2

4.lépés

Szabad erőforrások száma: (3,3,2)

5.lépés

Készlet: (3,3,2)
P1 kielégítő
Készlet: (5,3,2)
P3 kielégítő
Készlet: (7,4,3)
P3 kielégítő
Készlet: (7,5,3)
P3 kielégítő
Készlet: (10,5,5)
P3 kielégítő
Készlet: (10,5,7)

Processzek sorrendje: P1-P3-P0-P2-P4

6.lépés

A P1 lefutott, új készlet (3+2, 3+0, 2+0)
Készlet: (5,3,2)
A P3 lefutott, új készlet (5+2, 3+1, 2+1)
Készlet: (7,4,3)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MSGKEY 654311L
#define SIZE 2

struct msgbuf1 {
    long mtype;
    int  mtext[SIZE+1];
} message, *msgPointer;

int main() {
    int mymsg;
    key_t mykey;
    int myflag;
    int myreturn, mysize;
    int i;

    mykey = MSGKEY;
    myflag = 00666 | IPC_CREAT;
    mymsg = msgget(mykey, myflag);
    if (mymsg == -1) {
        perror("Hiba!");
        exit(-1);
    }
    printf("Sikeres letrehozas: %d, %x\n", mymsg, m
    msgPointer = &message;
    msgPointer->mtype = 0;

    for (i = 1; i < SIZE + 1; i++) {
        printf("Szamot pls: %d\n",i);
        scanf("%d", &(msgPointer->mtext[i]));
        printf("%d", msgPointer->mtext[i]);
    }
    mysize = sizeof(int) * (SIZE + 1);
    myreturn = msgsnd(mymsg, (struct msgbuf *)msgPo
    printf("Visszateresi ertek: %d\n", myreturn);
    printf("Uzenet: %s\n", msgPointer->mtext);

    exit(0);
}
```

```
mymsg);
```

```
pinter, mysize, myflag);
```