

## A feladat leírása

Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvassa meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon message queue(üzenetsoros) IPC mechanizmust, valamint a kimenet kerüljön egy másik fájlba. A kimeneti fájl struktúrája kötött! Példa a bemeneti és kimeneti fájl struktúrájára:

Bemeneti fájl:

i (Ez jelzi a számpárok darabszámát)

x y

Kimeneti fájl (Az x, y jelzi a bemeneti adatokat a z pedig a kimenet eredményét):

x y z

## Bemeneti fájl tartalma

5
21 24
30 10
16 20
66 99
39 33

## A feladat megoldása – Első program – „main függvény”

1. Szükséges header állományok deklarálása:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <sys/ipc.h>
#include <sys/msg.h>
```

2. Struktúra létrehozása a message queue-hoz:

```
struct mesg_buffer {
    long mesg_type;
    double a;
    double b;
    double m;
} message;
```

3. Egy két elemű tömb (a számpárnak) és egy int típusú változó (a számpárok számának, tudjuk, hogy egész szám) létrehozása

```
double input[2]; // a, b
int szamparok;
```

4. A bemenet.txt fájl megnyitása olvasásra

```
FILE *fp = fopen("bemenet.txt", "r");
if (fp < 0){
    perror("Hiba üres a file");
    exit(-1);
}
```

5. Első sor kiolvasása a fájlból és eltárolása a változóba

```
fscanf(fp, "%d", &szamparok);
printf("A fileban található szamparok... %d \n", szamparok);
```

6. Fájlban lévő számpárok eltárolása az input nevű tömbben, az „lnko” függvény hívása, a „kuldes” függvény hívása, az adatok kiírása

```
for(int i = 0; i < szamparok; i++){
    for(int k = 0; k < 2; k++){
        fscanf(fp, "%lf", &input[k]);
    }

    double m = lnko(input[0], input[1]);
    kuldes(input[0], input[1], m);
    printf("%d. szampar: a = %.2lf, b = %.2lf, m = %.2lf \n", i + 1, input[0], input[1], m);
}
```

7. Megnyitott fájl bezárása

```
fclose(fp);
return 0;
```

## A feladat megoldása – Első program – „lnko” függvény

1. A main függvény felett a prototípus létrehozása után a függvény létrehozása „main” alá

```
int lnko(double x, double y){  
  
    if (x == 0) {  
        return y;  
    }  
  
    while (y != 0) {  
        if (x > y)  
            x = x - y;  
        else  
            y = y - x;  
    }  
  
    return x;  
}
```

- legnagyobb közös osztó kiszámolása egyszerűbb egyenletekkel
- „y” a visszatérési érték, ha az „x” értéke 0
- amíg az „y” értéke nem egyenlő 0-val,  
 ha az „x” nagyobb, mint „y”, az „x” értéke „x – y” értéket vegye fel,  
 ellenkező esetben az „y” az „y – x” értéket vegye fel,  
 majd a ciklus visszatérési értéke „x”

## A feladat megoldása – Első program – „kuldes” függvény

1. A main függvény felett a prototípus létrehozása után a függvény létrehozása „main” és „lnko” alá

```
void kuldes(double a, double b, double m){  
  
    key_t key;  
    key = ftok("progfile", 65);  
    int msgid = msgget(key, 0666 | IPC_CREAT);  
    printf("msgid: %d\n", msgid);  
    message.mesg_type = 1;  
  
    message.a = a;  
    message.b = b;  
    message.m = m;  
  
    msgsnd(msgid, &message, sizeof(message), 0);  
    printf("LNKO: a = %.21f, b=%.21f, lnko=%.21f\n",a,b,m);  
}
```

- „key\_t key”
  - a kulcs az üzenetsorhoz
- „ftok()”
  - generál egy saját kulcsot
- „msgid”
  - a message queue azonosítója
- „mesg\_type”
  - az üzenet típusa
- „a, b, m”
  - a struktúra értékei a paraméterben kapott adatok
- „msgsnd()”
  - az üzenet elküldése

## A feladat megoldása – Második program – „main” függvény

1. Szükséges header állományok deklarálása:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <sys/ipc.h>
#include <sys/msg.h>
```

2. Struktúra létrehozása a message queue-hoz:

```
struct msg_buffer {
    long msg_type;
    double a;
    double b;
    double m;
}; message;
```

3. Kulcs létrehozása hibakereséssel

```
key_t key;
int msgid;
key = ftok("progfile", 65);

msgid = msgget(key, 0666 | IPC_CREAT);
printf("msgid: %d\n", msgid);
if(msgid == -1){
    perror("msgget");
    exit(1);
}
```

4. Ciklusban üzenetek fogadása ellenőrzéssel, „file” függvény hívása

```
for(;;){

    if(msgrcv(msgid, &message, sizeof(message), 1, 0) == -1){
        perror("msgrcv");
        exit(1);
    }

    file(message.a,message.b,message.m);
    printf("File-ba irt szoveg: \n");
    printf("LNKO: a = %.21f, b=%.21f, lnko=%.21f\n",message.a,message.b,message.m);
}
```

5. Üzenetsor megszüntetése „msgctl” segítségével

```
msgctl(msgid, IPC_RMID, NULL);
printf("Vege\n");

return 0;
```

## A feladat megoldása – Második program – „file” függvény

1. A main függvény felett a prototípus létrehozása után a függvény létrehozása „main” alá

```
void file(double a, double b, double m){  
  
    FILE *file_to_write = fopen("vege.txt","a");  
  
    if (file_to_write < 0){  
        perror("Hiba, ures a file");  
        exit(-1);  
    }  
  
    printf("Sikeres file kiiras\n");  
  
    fprintf(file_to_write,"LNKO: a = %.2lf, b=%.2lf, lnko=%.2lf\n",message.a,message.b,message.m);  
  
    fclose(file_to_write);  
}
```

- „FILE \*file\_to\_write”
  - fájl létrehozása
- „fopen()”
  - fájl megnyitása
- „if()” feltételes utasítás
  - fájl hibakeresése
- „fprintf()”
  - adatok beleírása a fájlba
- „fclose()”
  - fájl bezárása

## Kimeneti fájl tartalma

```
LNKO: a = 21.00, b=24.00, lnko=3.00
LNKO: a = 30.00, b=10.00, lnko=10.00
LNKO: a = 16.00, b=20.00, lnko=4.00
LNKO: a = 66.00, b=99.00, lnko=33.00
LNKO: a = 39.00, b=33.00, lnko=3.00
```

## Command Prompt (cmd)

```
hengsperger@jerry:~/BEADANDO_OS$ ./main.out
A fileban található szamparok... 5
msgid: 1638401
LNKO: a = 21.00, b=24.00, lnko=3.00
1. szampar: a = 21.00, b=24.00, m=3.00
msgid: 1638401
LNKO: a = 30.00, b=10.00, lnko=10.00
2. szampar: a = 30.00, b=10.00, m=10.00
msgid: 1638401
LNKO: a = 16.00, b=20.00, lnko=4.00
3. szampar: a = 16.00, b=20.00, m=4.00
msgid: 1638401
LNKO: a = 66.00, b=99.00, lnko=33.00
4. szampar: a = 66.00, b=99.00, m=33.00
msgid: 1638401
LNKO: a = 39.00, b=33.00, lnko=3.00
5. szampar: a = 39.00, b=33.00, m=3.00
hengsperger@jerry:~/BEADANDO_OS$ ./rec.out
msgid: 1638401
Sikeres file kiiras
File-ba irt szoveg:
LNKO: a = 21.00, b=24.00, lnko=3.00
Sikeres file kiiras
File-ba irt szoveg:
LNKO: a = 30.00, b=10.00, lnko=10.00
Sikeres file kiiras
File-ba irt szoveg:
LNKO: a = 16.00, b=20.00, lnko=4.00
Sikeres file kiiras
File-ba irt szoveg:
LNKO: a = 66.00, b=99.00, lnko=33.00
Sikeres file kiiras
File-ba irt szoveg:
LNKO: a = 39.00, b=33.00, lnko=3.00
```