JEGYZŐKÖNYV

**Operációs rendszerek BSc**

**2022. tavasz**

**Féléves feladat**

**Készítette: Dávid Rebeka**

**Neptunkód: EQ4B3D**

1. **Feladat: IPC mechanizmus**

**A feladat leírása:**

Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvasva 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!

Bemeneti fájl:

i (Ez jelenti a számpárok darabszámát)

x y

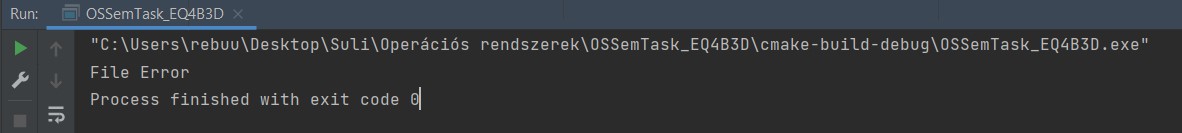
Kimeneti fájl (x, y jelzi a bemeneti adatokat, z pedig a kimeneti eredményt):

x y z

**A feladat elkészítése:**

#include<stdio.h>  
#include<stdlib.h>  
  
int gcd(int a, int b)  
{  
 if (a == 0)  
 return b;  
 return gcd(b % a, a);  
}  
  
int findGCD(int arr[], int n)  
{  
 int result = arr[0];  
 for (int i = 1; i < n; i++)  
 {  
 result = gcd(arr[i], result);  
  
 if(result == 1)  
 {  
 return 1;  
 }  
 }  
 return result;  
}  
  
int main()  
{  
  
 int numbers[50];  
 int i = 0;  
 FILE \*file;  
 if (file = fopen("input.txt", "r"))  
 {  
 while (fscanf(file, "%d ", &numbers[i]) != EOF)  
 {  
 i++;  
 }  
 fclose(file);  
 }  
 else  
 {  
 printf("File Error");  
 }  
 numbers[i] = '\0';  
  
 int n = sizeof(numbers) / sizeof(numbers[0]);  
 int result=findGCD(numbers,n);  
  
 FILE \*fp;  
 if (fp = fopen("output.txt", "w"))  
 {  
 fprintf(fp, "GCD is :%d\n", result);  
 fclose(fp);  
 }  
 else  
 {  
 printf("File Error");  
 }  
 return 0;  
}

**A futtatás eredménye:**

****

1. **Feladat: OS algoritmusok**

**A feladat leírása:**

Adott egy igény szerinti lapozást használó számítógépesrendszer, melynek futás közben egy processz számára a következő laphivatkozással lehet hivatkozni: 6,8,3,8,6,0,3,6,3,5,3,6.

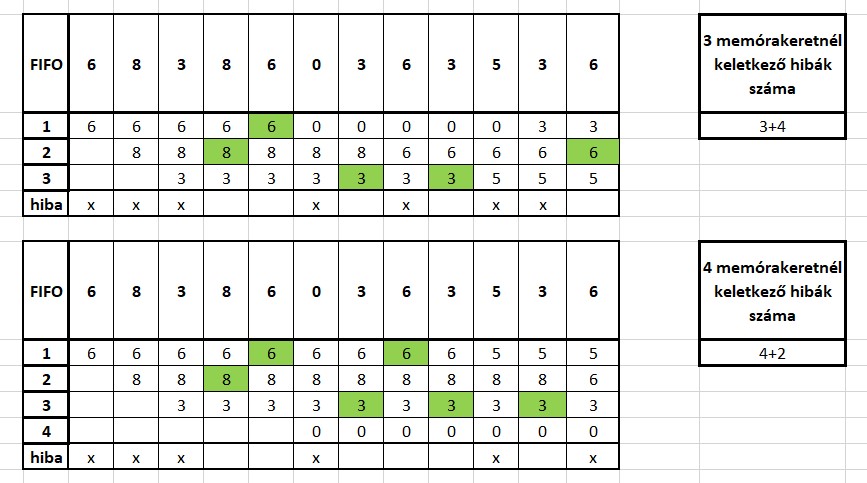
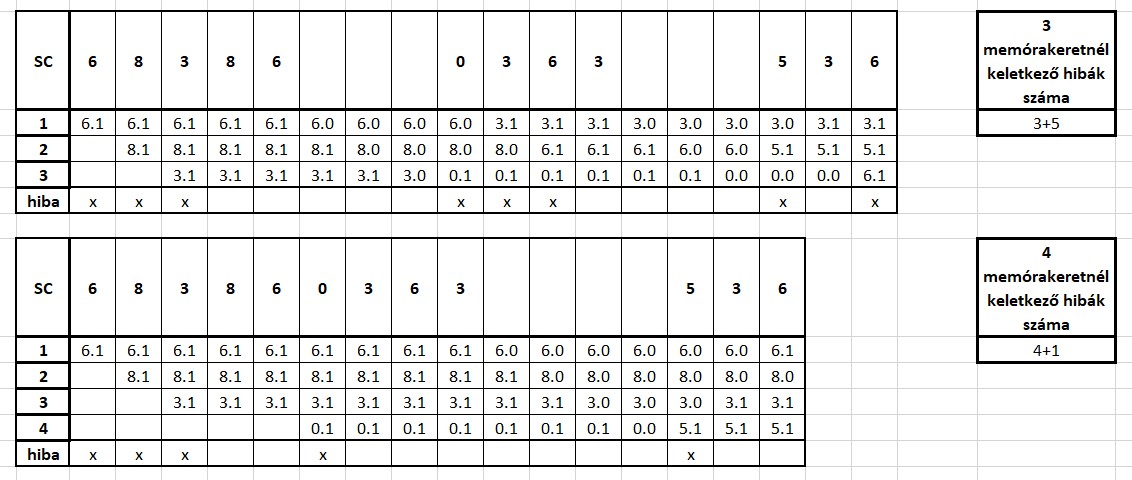
Memóriakeret (igényelt lapok): 3, ill. 4 memóriakeret.

Készítse el a laphivatkozások betöltését külön-külön táblázatba 3, ill. 4 memóriakeret esetén.

Mennyi laphiba keletkezik az alábbi algoritmusok esetén: FIFO, SC?

Hasonlítsa össze és magyarázza az eredményeket!

**A feladat elkészítése:**

* **FIFO:** Hibák szempontjából 4 memóriakeret esetén hatékonyabb
* **SC:** Hibák szempontjából 4 memóriakeret esetén hatékonyabb

**A FIFO és Second Chance algoritmust összehasonítva a legeredményesebb a Second Chance algoritmus 4 memóriakeret esetén, a legeredménytelenebb pedig a Second Chance 3 memóriakeret esetén.**