

Pràctica 1 de SIOP - Ixent Cornella Vilana

Exercici 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

short comprova(int a){

    if(a > 0)
        return 1;
    else
        return 0;
}

short enters(float a, float b){    //Fem un short ja que el resultat que
retorna la funcio sera 0, 1, en funcio de si a i b son enters.

    short result;
    int _a = a/1;                //Divdim els numeros per 1 per veure si son enters o no,
si no ho fossin, llavors el resultat es l'arrodoniment del
    int _b = b/1;                //float al enter mes proper.

    if(_a != a && _b != b){
        //printf ("Els numeros %f i %f no son enters.", a, b);            // si
els dos son iguals a l'operacio " x/1 ", no son enters
        result = 0;
    }
    else if (_a == a && _b != b){
        // printf ("El numero %d es enter, mentre que el numero %f no es enter.",
_a, b);
        result = 0;
    }
    else if (_a != a && _b == b){
        //printf ("El numero %f no es enter, mentre que el numero %d es enter.",
a, _b);
        result = 0;
    }
    else if(a > 0 && b > 0){
        printf ("Els dos numeros %d i %d son enters mes grans que 0.\n", _a,
_b);
        result = 1;
    }
    else{
        result = 0;
    }

    return result;
}

int main(int argc, char *argv[]){

    float num1, num2;            //Variables que introduira el usuari

    char resposta[10];           //Resposta del usuari si vol seguir el programa

    if(argc != 3){
```

```

        printf("ERROR: Has d'introduir exactament DOS numeros, el programa et
dirà si son enters o no.\n");
    }
    else{

        num1 = atof(argv[1]);
        num2 = atof(argv[2]);

        if(num1 <= 0 || num2 <= 0){
            printf("ERROR: Si us plau, introdueix unicament numeros (mes grans
que 0)\n");
        }
        else{

            short valid = enters(num1, num2);

            if(valid){

                if(fork() == 0){

                    short mes_gran = comprova(num2);
                    if (!mes_gran)
                        exit(-1);
                    else
                        exit(0);

                }
                int st;
                waitpid(-1, &st);
                if (st == 0){
                    printf("N Mes gran que 0.\n");
                }
                else printf("Error: N Mes petit que 0\n");

            }
            else{

                printf("ERROR: Introdueix NOMES nombres enters mes grans que
0.\n");

                exit(-1);
            }
        }
    }
}

```

/*
TT (Time thieves) equivaldria a un programa en C.
Un replicant és un fork (fill) d'un TT.

M i N son dues variables naturals.

Un missatge informatiu es un printf en el programa.

Finalitzar amb -1 equival a fer un exit amb codi -1.

*/

Exercici 2

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

short enters(float a, float b){    //Fem un short ja que el resultat que
retorna la funcio sera 0, 1, en funcio de si a i b son enters.

    short result;
    int _a = a/1;                //Divdim els numeros per 1 per veure si son enters o no,
si no ho fossin, llavors el resultat es l'arrodoniment del
    int _b = b/1;                //float al enter mes proper.

    if(_a != a && _b != b){
        //printf ("Els numeros %f i %f no son enters.", a, b);           // si
els dos son iguals a l'operacio " x/1 ", no son enters
        result = 0;
    }
    else if (_a == a && _b != b){
        // printf ("El numero %d es enter, mentre que el numero %f no es enter.",
_a, b);
        result = 0;
    }
    else if (_a != a && _b == b){
        //printf ("El numero %f no es enter, mentre que el numero %d es enter.",
a, _b);
        result = 0;
    }
    else if(a > 0 && b > 0){
        printf ("Els dos numeros %d i %d son enters mes grans que 0.\n", _a,
_b);
        result = 1;
    }
    else{
        result = 0;
    }

    return result;
}

int main(int argc, char *argv[]){

    float num1, num2;            //Variables que introduira el usuari

    char resposta[10];           //Resposta del usuari si vol seguir el programa

    if(argc != 3){
        printf("ERROR: Has d'introduir exactament DOS numeros, el programa et
dira si son enters o no.\n");
    }
    else{

        num1 = atof(argv[1]);
        num2 = atof(argv[2]);
    }
}
```

```

        if(num1 <= 0 || num2 <= 0){
            printf("ERROR: Si us plau, introduceix unicament numeros (mes grans
que 0)\n");
        }
        else{

            short valid = enters(num1, num2);

            if(valid){

                int valor;

                for(int i = 0; i < num1; i++){
                    int id = fork();
                    if(id == 0){

                        valor = num2;
                        printf("El meu valor es %d.\n", valor);
                        exit(0);

                    }
                }

            }
            else{

                printf("ERROR: Introdueix NOMES nombres enters mes grans que
0.\n");

                exit(-1);

            }

        }

    }

}

/*
Una mutació equivaldria a la comanda "execlp", cada fill tindria
la capacitat de canviar els parametres de la comanda execlp, i per tant,
canviar el seu codi, dit d'una altra manera, mutar.
*/

```

Exercici 3

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

short comprova(int a){

    if(a > 0)
        return 1;
    else
        return 0;
}

short enters(float a, float b){    //Fem un short ja que el resultat que
retorna la funcio sera 0, 1, en funcio de si a i b son enters.

    short result;
    int _a = a/1;                //Divdim els numeros per 1 per veure si son enters o no,
si no ho fossin, llavors el resultat es l'arrodoniment del
    int _b = b/1;                //float al enter mes proper.

    if(_a != a && _b != b){
        //printf ("Els numeros %f i %f no son enters.", a, b);            // si
els dos son iguals a l'operacio " x/1 ", no son enters
        result = 0;
    }
    else if (_a == a && _b != b){
        // printf ("El numero %d es enter, mentre que el numero %f no es enter.",
_a, b);
        result = 0;
    }
    else if (_a != a && _b == b){
        //printf ("El numero %f no es enter, mentre que el numero %d es enter.",
a, _b);
        result = 0;
    }
    else if(a > 0 && b > 0){
        printf ("Els dos numeros %d i %d son enters mes grans que 0.\n", _a,
_b);
        result = 1;
    }
    else{
        result = 0;
    }

    return result;
}

int main(int argc, char *argv[]){

    float num1, num2;            //Variables que introduira el usuari

    char resposta[10];           //Resposta del usuari si vol seguir el programa

    if(argc != 3){
```

```

        printf("ERROR: Has d'introduir exactament DOS numeros, el programa et
dira si son enters o no.\n");
    }
    else{

        num1 = atof(argv[1]);
        num2 = atof(argv[2]);

        if(num1 <= 0 || num2 <= 0){
            printf("ERROR: Si us plau, introdueix unicament numeros (mes grans
que 0)\n");
        }
        else{

            short valid = enters(num1, num2);

            if(valid){

                int M = num1;
                int N = num2;

                for(int i = 0; i < M; i++){
                    int id = fork();
                    if(id == 0){

                        short mes_gran = comprova(N);
                        if (!mes_gran)
                            exit(-1);
                        else
                            exit(0);

                    }
                }
                int st;
                waitpid(-1, &st);
                if (st == 0){
                    printf("Exit en la missio\n");
                }
                else printf("ERROR: El valor %d no es mes gran que 0. Missio
fallida.\n", N);

            }
            else{

                printf("ERROR: Introdueix NOMES nombres enters mes grans que
0.\n");

                exit(-1);
            }
        }
    }
}

```

```
}
```

```
/*
```

```
PSEUDOCODI:
```

```
if (input usuari no son dos enters)
```

```
    missatge d'error
```

```
altrament:
```

```
    crea M replicants, que comprovin que N sigui mes gran 0.
```

```
    if(N > 0)
```

```
        finalitza amb codi 0
```

```
    altrament:
```

```
        finalitza amb codi -1
```

```
    Esperar tots els fills...
```

```
    if (algun fill te codi -1)
```

```
        missatge d'error.
```

```
*/
```


Exercici 4

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "dummy.h"

short comprova(int a){

    if(a > 0)
        return 1;
    else
        return 0;
}

short enters(float a, float b){    //Fem un short ja que el resultat que
retorna la funcio sera 0, 1, en funcio de si a i b son enters.

    short result;
    int _a = a/1;                //Divdim els numeros per 1 per veure si son enters o no,
si no ho fossin, llavors el resultat es l'arrodoniment del
    int _b = b/1;                //float al enter mes proper.

    if(_a != a && _b != b){
        //printf ("Els numeros %f i %f no son enters.", a, b);            // si
els dos son iguals a l'operacio " x/1 ", no son enters
        result = 0;
    }
    else if (_a == a && _b != b){
        // printf ("El numero %d es enter, mentre que el numero %f no es enter.",
_a, b);
        result = 0;
    }
    else if (_a != a && _b == b){
        //printf ("El numero %f no es enter, mentre que el numero %d es enter.",
a, _b);
        result = 0;
    }
    else if(a > 0 && b > 0){
        printf ("Els dos numeros %d i %d son enters mes grans que 0.\n", _a,
_b);
        result = 1;
    }
    else{
        result = 0;
    }

    return result;
}

int main(int argc, char *argv[]){

    float num1, num2;            //Variables que introduira el usuari

    char resposta[10];          //Resposta del usuari si vol seguir el programa
```

```

    if(argc != 3){
        printf("ERROR: Has d'introduir exactament DOS numeros, el programa et
dirà si son enters o no.\n");
    }
    else{

        num1 = atof(argv[1]);
        num2 = atof(argv[2]);

        if(num1 <= 0 || num2 <= 0){
            printf("ERROR: Si us plau, introdueix unicament numeros (mes grans
que 0)\n");
        }
        else{

            short valid = enters(num1, num2);

            if(valid){

                int M = num1;
                int N = num2;

                for(int i = 0; i < M; i++){
                    int id = fork();
                    if(id == 0){

                        char buffer[N];

                        if (sizeof(buffer) < N)
                            exit(-1);
                        else
                            exit(0);

                        int final = 0;
                        // comprovacions inicials
                        dummy_ini (buffer, N);
                        while (!final)
                            final = dummy_cpt( buffer, N );
                        dummy_end();
                    }
                    int st;
                    waitpid(-1, &st);
                    if (st == 0){
                        printf("Exit en la missio\n");
                    }
                    else printf("ERROR: La memoria del contenidor es mes gran
que N. Missio fallida.\n");
                }

            }

            else{

                printf("ERROR: Introdueix NOMES nombres enters mes grans que

```

```
0.\n");  
  
        exit(-1);  
    }  
  
    }  
  
}  
/*  
    Robar temps equivaldria a guardar dins de variables "buffer"  
    blocs de N unitats de temps.  
*/
```