

# Examen de laboratori d'EC

30 de Maig de 2024 – De 12h a 13:55h

## INSTRUCCIONS IMPORTANTS:

### Baixa els fitxers plantilla de la web d'exàmens

1. La imatge Linux que cal usar hauria d'estar precarregada ja quan comences a fer l'examen. Si no ho estigués, cal usar la imatge "**Linux**" que apareix al menú de possibles imatges.
2. Entra al teu compte personal amb el teu nom d'usuari d'estudiant.
3. Obre el navegador a la pàgina: <https://examens.fib.upc.edu>
4. Clica *Accedir a l'aplicació d'exàmens*, i autentifica't de nou amb el teu nom d'usuari d'estudiant.
5. Busca la pràctica *Examen de Laboratori d'EC - Quadrimestre primavera 2023/24 – Subgrup 11*
6. Obre el fitxer adjunt **examlab-2324q2-subgrup11.zip**, i extrau els fitxers a la teva carpeta. Comença l'examen.

### Resol l'examen

1. Posa nom i cognoms a la capçalera dels fitxers següents i que s'han de lliurar de forma individual:  
**matrius.s, subrutines.s, respostes\_caches.txt**
2. No canviïs el nom dels fitxers.
3. Pots pujar cada fitxer a la web múltiples cops, no esperis a l'últim minut.
4. L'avaluació de cada apartat tindrà en compte principalment el seu funcionament correcte. Es practica-  
ran diverses comprovacions, a més a més de la que suggereix el propi enunciat.
5. Per executar el simulador escriviu la següent comanda en una finestra de terminal:  

```
java -jar /assig/ec/Mars/Mars.jar &
```

### Puja les solucions a la web d'exàmens

1. Busca al final de la pàgina web el quadre *Lliurar una nova pràctica*, i verifica que la llista desplegable té seleccionada la pràctica: *Examen de laboratori d'EC - Quadrimestre primavera 2023/24 – Subgrup 11*
2. Clica el botó *Browse* i puja d'un en un els fitxers a lliurar amb la solució:  
**matrius.s, subrutines.s, respostes\_caches.txt**
3. Verifica que cada fitxer que has pujat no és un fitxer plantilla buit (comprovant la mida en bytes!) i que els noms són els corresponents a aquest torn d'examen.
4. Surt de la sessió que tinguis oberta però **NO APAGUIS EL PC**.

### Problema 1. (3,5 punts)

Donat el següent codi en C, escriu al fitxer **matrius.s** el programa equivalent en assemblador de MIPS.

```
unsigned short dIN[6][6] = {1, 2, 3, 4, 5, 6,
                             2, 3, 4, 5, 6, 7,
                             3, 4, 5, 6, 7, 8,
                             4, 5, 6, 7, 8, 9,
                             5, 6, 7, 8, 9, 10,
                             6, 7, 8, 9, 10, 11};

int dOUT[4][4]; // Inicialitzada a 0

main(){
    for (int i=1; i<=4; i++ )
        for (int j=1; j<=4; j++ )
            dOUT[i-1][j-1] = dIN[i+1][j] + dIN[i][j-1] + dIN[i][j+1];
}
```

Tradueix al fitxer **matrius.s** el codi de main. Les variables globals ja estan declarades i inicialitzades. Has d'utilitzar la tècnica d'accés seqüencial per accedir a la matriu dOUT.

Comprova que al final de l'execució del programa la matriu global dOUT conté els valors següents:

```
int dOUT[4][4] = { 10, 13, 16, 19,
                   13, 16, 19, 22,
                   16, 19, 22, 25,
                   19, 22, 25, 28};
```

## Problema 2. (3,5 punts)

Tradueix a ensamblador MIPS en el fitxer **subrutines.s** el codi de la funció **hanoi** seguint les regles definides a l'ABI de MIPS. Recorda que no pots modificar el codi de la resta de funcions, ni afegir variables globals.

```
int MOVs;

void mover(char A, char B);

int hanoi(char A, char B, char tmp, int N) {
    int sum, c1, c2;

    if (N == 1) {
        mover(A, B);
        sum = 1;
    }
    else {
        c1 = hanoi(A, tmp, B, N-1);
        mover(A, B);
        c2 = hanoi(tmp, B, A, N-1);
        sum = c1 + c2 + 1;
    }
    return sum;
}

void main() {
    MOVs = hanoi('A', 'B', 'X', 4);
}
```

Comprova que al final del programa, la variable **MOVs** val **15**, i en la finestra de I/O apareix:

```
move A --> X
move A --> B
move X --> B
move A --> X
move B --> A
move B --> X
move A --> X
move A --> B
move X --> B
move X --> A
move B --> A
move X --> B
move A --> X
move A --> B
move X --> B
```

### Problema 3. (3 punts)

Considera un sistema format per un processador MIPS, una memòria principal (MP) i una memòria cache de dades (MC), amb la següent configuració:

- correspondència directa
- mida del bloc: 16 bytes (4 words)
- capacitat: 64 bytes (4 blocs)
- escriptura immediata amb assignació

Considera el següent programa en C (que s'ha traduït a MIPS en el fitxer **caches.s**):

```
int A[24];
int MAXpar[12];

void main() {

    for (int i=0; i<24; i=i+2)           // i es guarda a un registre
        if (A[i] > A[i+1])
            MAXpar[i/2] = A[i];
        else
            MAXpar[i/2] = A[i+1];
}
```

Suposa que les variables globals **A** i **MAXpar** s'emmagatzemen en posicions consecutives a partir de l'adreça 0x10010000 i que la MC està inicialment buida. Respon (al fitxer **respostes\_caches.txt**):

- (1 punt) Quin és el percentatge d'encert d'aquests programa en aquesta cache? Quantes fallades es produeixen per accessos a **A** i quantes fallades es produeixen per accessos a **MAXpar**?
- (1 punt) Quin és el percentatge d'encert d'aquests programa en aquesta cache si, sense canviar la mida de la cache, fem que sigui associativa de dues vies? Quantes fallades es produeixen per accessos a **A** i quantes fallades es produeixen per accessos a **MAXpar**?
- (0,5 punt) Justifica els canvis en el nombre de fallades que es produeixen a l'apartat b), i relaciona-ho amb el tipus de localitat (espacial o temporal) que té cadascuna de les variables.
- (0,5 punt) Justifica si seria possible reduir més la taxa de fallades de la memòria cache de l'apartat (b) canviant alguna de les seves característiques (nombre de vies, capacitat, mida de bloc,...). Explica quina característica canviaries i perquè.