



Pre-Final de PBN
Enginyeria de Sistemes TIC

15 de juny de 2012

150 MINUTS

COGNOMS:

NOM:

GRUP de LAB:

Exercici 1 [1 punt]. Suposem que es defineixen les següents variables:

```
int i,*j;  
int t[10], u[20];
```

Quina de les següents sentències és errònia?

- j = &i; *j = 3;
- j = t; *(j+2) = 14;
- u = t; u[2] = 7;
- u[20] = (int)(&i); *((int *)u[20]) = 4;

Exercici 2 [1 punt]. Considereu la següent capçalera de funció:

```
int * f1(int t[], int *const k);
```

Quina de les següents afirmacions és correcta?

- La capçalera és errònia ja que t no indica el nombre de cel·les.
- El paràmetre k no es pot modificar en el cos de la funció.
- Les modificacions que la funció faci a t no afectaran al corresponent argument.
- La capçalera és errònia ja que una funció no pot tornar un apuntador.

Exercici 3 [1 punt]. Considereu el següent fragment de codi d'un mòdul C:

```
#define A 23  
const int B = 23;  
static const int C = 23;
```

Quina de les següents afirmacions sobre la taula de símbols del mòdul és certa?

- La taula contindrà el símbol A.
- La taula contindrà el símbol B.
- Cap dels símbols A, B i C sortirà a la taula de símbols.
- El símbol C sempre sortirà a la taula de símbols.

Exercici 4 [1 punt]. Considereu el següent fragment de codi C:

```
#define EXP (12 & 0x15 + 67)

i = EXP;
```

Quina de les següents afirmacions és certa?

- La complicació de l'expressió EXP no afecta al temps d'execució per que la calcula el pre-processor.
- La complicació de l'expressió EXP no afecta al temps d'execució per que la simplifica l'optimitzador.
- La complicació de l'expressió EXP afecta al temps d'execució.
- No es pot saber si la complicació de l'expressió EXP afecta o no al temps d'execució.

Exercici 5 [1 punt]. Supposeu que un `makefile` conté les següents dependències:

```
a.o: a.c a.h b.h
b.o: b.c b.h c.h
c.o: c.c c.h
```

Quina de les següents afirmacions és certa?

- Si modifiquem `c.h` caldrà recompilar tots els mòduls.
- Si recompilem `a.c` també cal recompilar `b.c`.
- Per forçar una recompilació de tots els mòduls podem modificar `c.h` i `b.h`.
- Els objectes mai depenen dels headers sinó únicament dels fonts.

Exercici 6 [1 punt]. Definiu el concepte de símbol no resol't.

Exercici 7 [2 punts]. Situeu-vos en el context de les pràctiques. Imagineu que volem transmetre el senyal morse de forma diferent: tant punts com ratlles tenen la mateixa durada (un DOT exactament) però es representen amb un so de freqüència diferent: els punts 3.1 kHz i les ratlles 10 kHz. Feu un redisseny del projecte per encabir aquesta modificació: quins mòduls es veurien afectats? Com canviarien les seves API's? Com canviarien les seves implementacions?

Exercici 8 [2 punts]. Observa la implementació de l'operació enqueue del mòdul queue:

```

void queue_enqueue(queue_t *const q, uint8_t v) {
    if (!queue_is_full(q)) {
        q->t[ q->rear ] = v;
        q->rear = inc(q->rear);
    }
}

```

En cas que una rutina d'interrupció també interaccionés amb la cua, quina zona de l'operació enqueue caldria considerar-la en exclusió mútua i per quina raó?

Exercici 9 [5 punts]. L'objectiu d'aquest problema és dissenyar i implementar un un sistema d'"home mort". Els sistemes d'home mort són uns sistemes usats en els ferrocarrils per a prevenir accidents causats per la mort o defalliment del maquinista. El seu funcionament és el següent:

Cada un cert temps arbitrari es dispara una alarma en forma de llum a la cabina del conductor. Si el maquinista no prem un botó en menys d'un temps δ des de que es dispara el senyal, el comboi s'atura automàticament. Si el conductor prem el botó dins del temps previst el llum s'apaga i no té cap més conseqüència: el tren continua funcionant.

Es demana que, assumint que teniu disponibles tots els mòduls del projecte, implementeu un programa principal que faci aquesta feina. Per simplificar-ho suposeu que, en comptes d'un comboi ferroviari, teniu un LED que fa aquest paper: quan està encès el tren està en marxa,

quan està apagat el tren està aturat. Inicialment el tren està en marxa i el sistema d'home mort funcionant.

Per saber si algú ha polsat el polsador d'home mort useu un port d'entrada del microcontrolador. També us caldrà una font de dades aleatòries. A tal efecte podeu assumir que existeix un mòdul que ja està implementat amb aquesta funcionalitat.

Es valorarà la capacitat d'usar els mòduls disponibles de manera eficaç, la bondat de l'estructura del programa resultant i la claredat en el desenvolupament de la resposta. Si ho creieu necessari podeu definir nous mòduls, que també haureu d'implementar, per tal d'estructurar correctament la solució.