Solució examen codificació i lògica binària (II)

	DECIMAL	HEXADECIMAL	BINARI
1	**92**		
2		**F7**	
3			**1001·1011**
4	**27**		
5			**1010·1010**
6	**31**		

	Decimal		Binari
	53		
-	**15**	**+**	
	38		

	Decimal		Binari
	31		
-	**27**	**+**	

(2 pt) Preguntes:

He escrit un programa en Java que suma 1000 cèntims un a un fent servir punt flotant. El resultat és 9,999999999999831.

1. **Per què penses que he obtingut aquest valor**? El 0.1 no és representable en binari.

2. El meu company em diu que he de fer servir variables de tipus double, perquè tenen més precisió. **Penses que té raó**? No. En realitat aquest valor s'obté fent servir

variables de tipus double; si fem servir variables de tipus float l’error és encara més gran.

3. **Quina solució podries plantejar tu**? **Et sembla bona la solució del meu company**?

   No és un problema de precisió; amb una gran precisió només s’amaga el problema fent l’error més petit, però si s’acumulen molts errors l’error pot tornar a fer-se gran.

   Solucions:

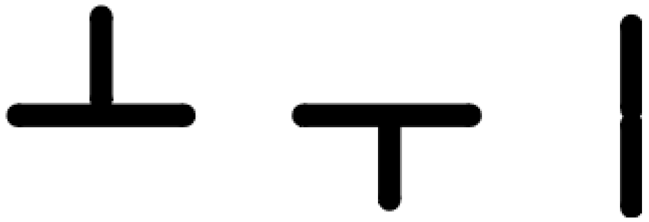
   1) Fer servir BCD. En Java existeix una llibreria BCD, però no està gaire pensada per fer càlculs massius i en Java és ineficient (en C# o en Python es pot fer servir BCD sense tant de cost com en Java, però pensem que operar en BCD sempre té un cost).
   2) Arrodonir. Per arrodonir a 2 decimals en Java hauriem de fer servir una cosa semblant a aquesta:

      ```
      a = Math.round(a*100)/100;
      ```

   3) Convertir els cèntims en unitats per fer els càlculs de centaus i dividir finalment per cent. L’error encara es donarà, però d’una única operació i serà quasi inapreciable; si a més a més arrodonim fent servir la solució anterior quedarà un resultat net, el mínim error acumulat i el mínim cost addicional.

(6 pt) Problema:

En el planeta Tarna els nadius compten en base 3, fent servir 3 símbols:



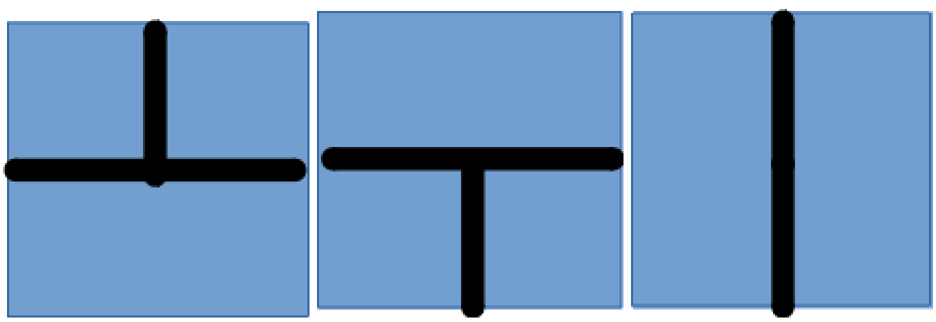
(E0) (E1) (E2)

Volem construir un display fent servir 3 entrades (E0, E1 i E2), una per a cada valor, de forma que només una de les tres pot estar activa simultàniament, mostrant llavors el símbol que correspon a l'entrada. Quan totes 3 entrades estan inactives el display queda apagat.

Els nostres enginyers han dit que el display es pot construir fent servir 3 barres que s'il·luminen quan reben un senyal:

1. Barra horitzontal, centrada verticalment: B0
2. Mitja barra superior, centrada horitzontalment: SB1
3. Mitja barra inferior, centrada horitzontalment: SB2

Amb B0, SB1 i SB2 es poden construir els tres símbols:



E0 = 1	E0 = 0	E0 = 0
E1 = 0	E1 = 1	E1 = 0
E2 = 0	E2 = 0	E2 = 1

Preguntes:

1. Fes les taules de veritat per a cada barra (B0, SB1, SB2).

A) Establint la condició nosaltres, mirant el comportament. Així podriem pensar que:

B0 = E0 OR E1
SB1 = E0 OR E2
SB2 = E1 OR E2

B) La expressió lògica la podem construir primer, o bé, si pensem de forma més visual, podem construir primer la taula de veritat i després concloure quines són les expressions, però ens calen totes dues; la taula es demana, i a més a més la taula **estudia tots els casos**, i ens garanteix que el resultat és correcte. Les expressions ens calen pel següent punt.

E0	E1	E2	B0	SB1	SB2
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Comprovem **els casos en verd**, que **són els que han de funcionar**, i efectivament la taula és correcte.

ANNEX: En realitat **la condició la podem fer més exigent**. Aquest punt és important explicar-ho, perquè és la forma de fer tots els casos, que no havíem vist a classe (però no calia per l'examen). Potser algú s'ha embolicat volent arribar a aquesta solució:

B0 = (E0 AND NOT E1 AND NOT E2) OR (E1 AND NOT E0 AND NOT E2)
= NOT E2 AND ((E0 AND NOT E1) OR (E1 AND NOT E0))
= NOT E2 AND (E0 XOR E1)

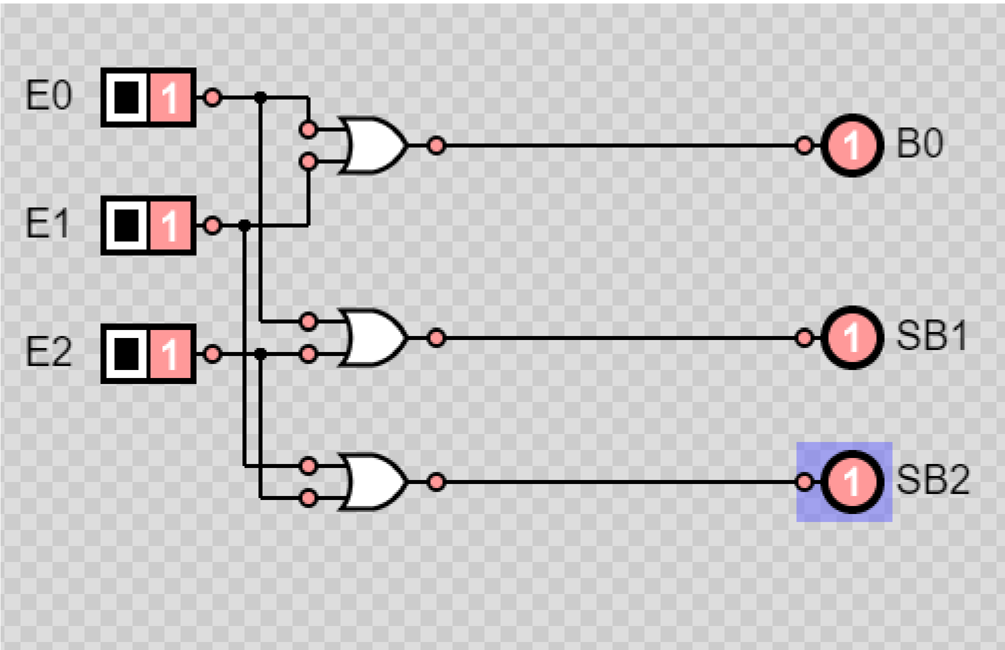
SB1 = (E0 AND NOT E1 AND NOT E2) OR (E2 AND NOT E0 AND NOT E1)
= NOT E1 AND ((E0 AND NOT E2) OR (E2 AND NOT E0))
= NOT E1 AND (E0 XOR E2)

SB2 = (E1 AND NOT E0 AND NOT E2) OR (E2 AND NOT E0 AND NOT E1)
= NOT E0 AND ((E1 AND NOT E2) OR (E2 AND NOT E1))
= NOT E0 AND (E1 XOR E2)

Però en realitat, en aquest cas **no es guanya res complicant l'expressió**, perquè ens han dit que NOT (E0 AND E1) AND NOT (E1 AND E2) AND NOT (E0 AND E2), és a dir, que no es donen dues senyals d'entrada simultàniament.

Naturalment el més senzill és el millor.

2. Dibuixa els 3 circuits, un per a cada barra (B0, SB1, SB2) amb portes lògiques.



3. Només per pujar nota, si et sobra temps: quan s'activen dues entrades, cosa que no hauria de passar mai, s'han d'encendre totes tres barres per assenyalar l'error. Creus que serà difícil fer els canvis per incloure aquest requeriment? Afegeix-ho al que has fet en les preguntes 1 i 2.

Quan s'activen dues entrades és:

(E0 AND E1) OR (E1 AND E2) OR (E0 AND E2)

Òbviament, el cas d'activar-se totes tres ja queda inclòs. Només cal que amb aquest senyal totes tres barres s'encenguin.

B0 = E0 OR E1 OR (E0 AND E1) OR (E1 AND E2) OR (E0 AND E2)
SB1 = E0 OR E2 OR (E0 AND E1) OR (E1 AND E2) OR (E0 AND E2)
SB2 = E1 OR E2 OR (E0 AND E1) OR (E1 AND E2) OR (E0 AND E2)

Arribant a aquest punt tenim:

B0 = E0 OR E1 OR (E0 AND E1) OR (E1 AND E2) OR (E0 AND E2)

Algú podria adonar-se que E0 OR E1 inclou (E0 AND E1), efectivament perquè ja hem vist que ens hem desentès dels casos incorrectes, i aquest OR en realitat hauria de ser un XOR.... però arribem pel camí fàcil, que és el millor, així que podem simplificar:

A OR B OR (A AND B) = A OR B

B0 = E0 OR E1 OR (E1 AND E2) OR (E0 AND E2)
SB1 = E0 OR E2 OR (E0 AND E1) OR (E1 AND E2)
SB2 = E1 OR E2 OR (E0 AND E1) OR (E0 AND E2)

Traient factor comú:

B0 = E0 OR E1 OR (E2 AND (E0 OR E1))
SB1 = E0 OR E2 OR (E1 AND (E0 OR E2))
SB2 = E1 OR E2 OR (E0 AND (E1 OR E2))

Tabla de Verdad

E0	E1	E2	B0	SB1	SB2
0	0	0	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
1	1	0	1	1	1
0	0	1	0	1	1
1	0	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1