Un laboratori d'investigació cultiva una colònia de bacteris dins d'una àrea que es pot considerar com una superfície quadriculada de dimensió 30 x 30. Cada casella pot ser buida o contenir un bacteri. A partir de la seva configuració inicial, la colònia evoluciona generació rere generació segons unes lleis genètiques que tot seguit es descriuen i que depenen del nombre de veïns que té cada casella:

- **Naixement:** tota casella buida amb exactament tres veïns tindrà un naixement la propera generació.
- **Mort per solitud:** tot bacteri que ocupa una casella amb 0 o 1 veïns morirà per solitud la propera generació.
- **Supervivència:** tot bacteri que ocupa una casella amb 2 o 3 veïns sobreviurà la propera generació.
- **Mort per asfixia:** tot bacteri que ocupa una casella amb més de 3 veïns morirà per asfixia la següent generació.

Noteu que cada bacteri té com a molt 8 veïns i que en el cas dels bacteris residents a les vores de la quadrícula el nombre de veïns és menor.
Es considera que la transició entre generacions és simultània en totes les caselles de la colònia.
Es demana dissenyar un programa que simuli l'evolució de la colònia de bacteris i determini, a partir d'una situació inicial aleatòria, quantes iteracions es necessiten per tal que la colònia arribi a una situació estable.

El programa serà semblant al següent:

Programa bacteris
    Inicialitzar variables
    Generar generació inicial
    Mostrar generació inicial
    Mentre no situació estable **fer**
        Generar següent generació
        situació estable =(generació actual = =següent generació)
        generació actual = següent generació
    fMentre
    Mostrar generació final
    Mostrar nombre d'iteracions realitzades
fPrograma

Aclariments sobre algunes funcions

El procediment **Generar generació inicial** ha de crear una colònia de forma aleatòria. La forma més senzilla en és omplir la matriu amb valors aleatoris. Random r =new Random(); r.nextInt(2) i us retornarà valors que seran o bé 0 o bé 1 (podeu considerar per exemple que 1 representa que hi ha un bacteri i 0 que no).

El procediment **Mostrar generació inicial** simplement traurà per pantalla l'estat actual. No cal fer virgueries en la presentació, simplement que s'entengui el que hi ha. El procediment **Mostrar generació final** és exactament el mateix, simplement canviarà el paràmetre que li passem.

El procediment **Generar següent generació** ens crea la nova generació seguint les regles explicades abans. Òbviament per poder-ho fer haurà de cridar a una funció **veïns** que ens indiqui per a cada posició de la matriu, quants veïns n'hi ha.

Considerem que la situació és estable quan després de crear una nova generació ja no hi ha canvis a la colònia.
Aquest sistema genètic funciona, però per possibles errors de programació podria ser que entrés en un bucle infinit, per això fins que no esteu segurs que el programa funciona i acaba bé podeu posar també com a condició del bucle que no doni més de 500 voltes. També és cert que en algun cas pot entrar en un cicle de dues voltes i per tant serà millor que poseu sempre aquesta condició.
Si no teniu clar on falla el programa, es pot dintre del bucle visualitzar cada vegada el tauler, però això ralentitzarà molt l'execució. Si ho voleu provar d'aquesta manera, feu-lo amb matrius petites (6 x 6 o així). Quant tingueu clar que funciona, torneu a donar la mida inicial.

NOTES: El programa ha d'estar perfectament estructurat i la filosofia és l'explicada abans. No cal que sigui exactament així però penso que no pot canviar molt l'estructura.
Tot ha de fer-se mitjançant funcions (potser **Inicialitzar variables** no cal) i heu de pensar que el programa principal no quedarà gaire més llarg que el programa de la pàgina anterior.
Cal comentar el codi.