Consensus probabiliste (Ben-Or 1983)

On considère un système à n processus {1,2, .., n} avec au maximum f fautes franches, n>2f. Le système est asynchrone (pas de borne sur les délais de transmission et de traitements des messages). Les processus ont accès à un générateur de nombre aléatoire (r.n.g) qui retourne uniformément soit 0 soit 1.

L'algorithme suivant est exécuté par chaque processus p. Cet algorithme réalise un consensus binaire (entre deux valeurs 0 ou 1)

```
1.  procedure consensus(vp) {vp is the initial value of process p}
2.    x ← vp {x is p's current estimate of the decision value}
3.    k ← 0
4.    while true do
5.       k ← k + 1 {k is the current phase number}
6.       send (R, k, x) to all processes
7.       wait for messages of the form (R, k, ∗) from n − f processes
8.       if received more than n/2 (R, k, v) with the same v
9.       then send (P, k, v) to all processes
10.      else send (P, k, ?) to all processes
11.      wait for messages of the form (P, k, ∗) from n − f processes
12.      if received at least f + 1 (P, k, v) with the same v ≠ ?
13.      then decide(v)
14.      if at least one (P, k, v) with v ≠ ?
15.      then x ← v
16.      else x ← 0 or 1 randomly {query r.n.g.}
```

Question 1

Lors d'une ronde k, est-il possible qu'un processus propose 0 et un autre 1 ?

Question 2

Si un processus p décide v à la ronde k, montrez que tout processus q qui commence à la phase k+1 positionne sa variable x_q à v et décide v à la fin de la phase k+1

Question 3

Dans quel cas une valeur aléatoire est choisie ?

Montrez que cet algorithme termine avec une probabilité de 1.