Exercice 1 : On considère deux pièces de monnaie. La première tombe sur `Pile` avec probabilité $p_1 = \frac{1}{3}$ tandis que la seconde est équilibrée. On lance indéfiniment chacune de ces deux pièces. On pose :

- X_1 la v.a.r. égale au rang du premier lancer qui a donné `Pile` avec la première pièce ;
- X_2 la v.a.r. égale au rang du premier lancer qui a donné `Pile` avec la deuxième pièce ;
- $Y = \min(X_1, X_2)$ et $Z = \max(X_1, X_2)$.

Compléter la fonction suivante pour qu'elle renvoie une liste `[Y,Z]` contenant une simulation du couple (Y, Z), sans utiliser les fonctions `min` et `max` de **Python** :

```
def simulYZ():
    X1, X2 = ____________________________________________
    if ___________:
        ____________________
    else:
        ____________________
```

Exercice 2 : On considère une pièce de monnaie équilibrée, que l'on lance n fois ($n \geq 2$). On pose :

- X la variable aléatoire égale au rang du premier changement de résultat ou égale à 0 si tous les lancers donnent le même résultat ;
- Y la variable aléatoire de Bernoulli qui vaut 1 si on a obtenu au moins deux `Face` lors des n lancers et 0 sinon.

Exemple avec $n = 6$: si les lancers ont donné

`Face, Face, Face, Pile, Pile, Face`

alors X prend la valeur 4 et Y prend la valeur 1.

Exemple avec $n = 7$: si les lancers ont donné

`Pile, Pile, Face, Pile, Pile, Pile, Pile`

alors X prend la valeur 3 et Y prend la valeur 0.

Compléter la fonction suivante (dans laquelle les `Face` sont codés par 1) pour qu'elle renvoie une simulation du couple (X, Y) :

```
def simulXY(n):
    lancers = [________________________ for k in range(n)]
    if sum(lancers) ________:
        Y = 1
    else:
        Y = 0
    for k in range(1,n) :
        if lancers[k] != ________________:
            return __________
    return _________
```