Exercice 1 :

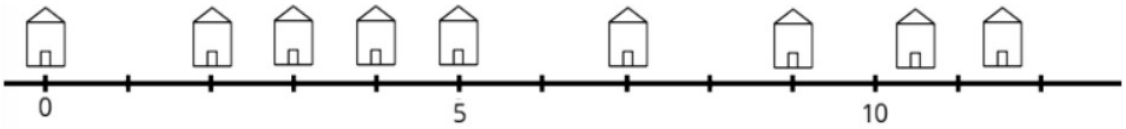
1. Création des deux instances :

```
m1 = Maison(1)
m2 = Maison(3.5)
```

2. Création de l'antenne :

```
a = Antenne(2.5, 1)
```

3. Schéma :



4.

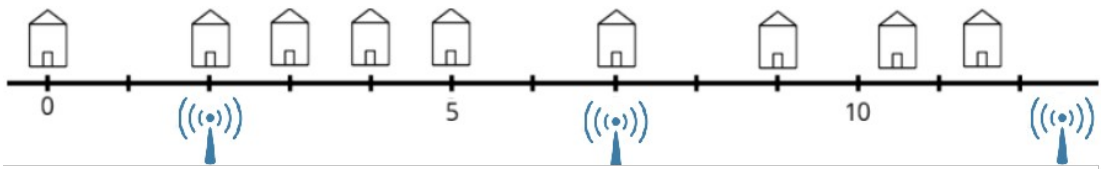
```
def creation_rue(pos):
    pos.sort()
    maisons = []
    for p in pos:
        m = Maison(p)
        maisons.append(m)
    return maisons
```

5.

```
# Méthode à ajouter dans la classe Antenne
def couvre(self, maison):
    distance = abs(self.get_pos_antenne() - maison.get_pos_maison())
    return distance <= self.get_rayon()
```

6. Cette suite d'instruction renvoie les positions des antennes selon la stratégie 1 qui consiste à place une antenne sur la première maison puis vérifie si la maison suivante est couverte par celle-ci, sinon ajoute une antenne au niveau de la maison suivante et ainsi de suite.
Le résultat est : [0, 3, 7, 10.5]

7.



8.

```
def strategie_2(maisons, rayon):
    antennes = [Antenne(maisons[0].get_pos_maison()+rayon, rayon)]
    for m in maisons[1:]:
        if not antennes[-1].couvre(m):
            antennes.append(Antenne(m.get_pos_maison()+rayon, rayon))
    return antennes
```

9. Quelque soit la stratégie, il faut parcourir la liste des maisons (taille du problème : n) ainsi le coût est linéaire : O(n).