

# Test pile

```
In [ ]: def creer_pile_vide():
        return []
```

```
In [ ]: def est_vide(p):
        return p==[]
```

```
In [ ]: def empiler(p,element):
        p.append(element)
```

```
In [ ]: def depiler(p):
        return p.pop()
```

```
In [ ]: def sommet(p):
        return p[-1]
```

```
In [ ]: def taille(p):
        return len(p)
```

```
In [ ]: def reduire_triplet_au_sommet(p):
        a = depiler(p)
        b = depiler(p)
        c = sommet(p)
        if a % 2 != c%2 :
            empiler(p, b)
        empiler(p, a)
```

```
In [ ]: def parcourir_pile_en_reduisant(p):
        q = creer_pile_vide()
        np = p.copy()
        while taille(np) >= 3:
            reduire_triplet_au_sommet(np)
            e = depiler(np)
            empiler(q, e)
        while not est_vide(q):
            e = depiler(q)
            empiler(np,e)
        return np
```

```
In [ ]: def jouer(p):
        q = parcourir_pile_en_reduisant(p)
        if taille(q)==taille(p) :
```

```
        return p
    else:
        return jouer(q)
```

```
In [ ]: tp = [2,4,7,8,9,4]
        tp1 = parcourir_pile_en_reduisant(tp)
        print(tp1)
        tp2 = parcourir_pile_en_reduisant(tp1)
        print(tp2)
        tp3 = parcourir_pile_en_reduisant(tp2)
        print(tp3)
```

```
In [ ]: print(jouer(tp))
```