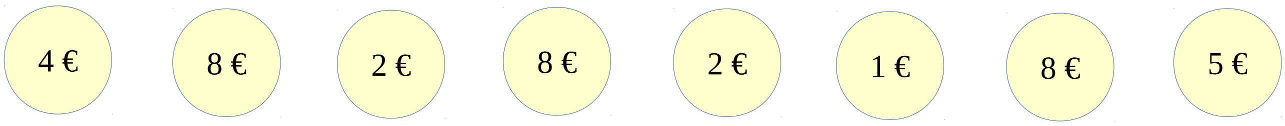
LE JEU DES PIECES DE MONNAIES



Considérons le jeu suivant , se jouant à 2 :

- il y a n pièces de monnaies placées sur une seule rangée, n est pair
- à tour de rôle, chaque joueur prend une pièce à une extrémité
- le jeu s'arrête quand toutes les pièces ont été prises

Le but de l'exercice est de déterminer **le montant minimal garanti** qu'un joueur peut gagner, en jouant le mieux possible.

Nous proposons de modéliser le problème de la façon suivante :

- il y a n pièces
- on note v_i la valeur de la i-ème pièce, pour i allant de 1 à n

Questions

1/ Faites 4 parties avec votre voisin : 2 parties où vous commencer, 2 parties votre voisin commence. Notez le nombre de fois où celui qui commence gagne.

2/Soit **V(i,j)** le meilleur montant que l'on peut assurément amasser s'il reste les pièces

(i,i+1,i+2, … , j-1, j)

(avec $i \leq j$).

Trouver une formule donnant la valeur de V(i,i)
Trouver une formule donnant la valeur de V(i,i+1)
Trouver une formulation récursive de V(i,j) pour les autres cas, en expliquant pourquoi.

3/ Calculer le tableau **V(i,j)** complet pour l'exemple proposé.

4/ Dans notre exemple, quel est le montant minimal garanti pour le joueur qui commence ?

5/Ecrivez une implémentation d'un algorithme calculant V(i,j)

Question 2

Cas de base

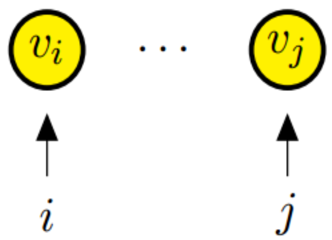
- Soit $V(i,j)$ le **meilleur montant** qu'on peut assurément amasser s'il reste les pièces

$$i, i+1, i+2, \ldots, j-1, j.$$

- Comme on l'a précisé plus tôt, il est **facile** de calculer
  - $V(i,i)$, pour $i = 1, 2, \ldots, n$;
  - $V(i,i+1)$, pour $i = 1, 2, \ldots, n-1$.

Formulation récursive

- De façon générale on doit calculer le **meilleur résultat** qu'on peut espérer en choisissant une pièce en position i ou en position j :



- Il y a **deux possibilités** :
  - On choisit **la pièce** i et alors on regarde le **pire scénario** qui peut survenir ;
  - On choisit **la pièce** j et alors on regarde le **pire scénario** qui peut survenir ;

Formulation récursive (suite)

- Bref, on a la **formule** suivante :

$$V(i,j) \quad = \quad \max\{ \quad \min\{V(i+2,j), V(i+1,j-1)\} + v_i, \\ \min\{V(i,j-2), V(i+1,j-1)\} + v_j \quad \}.$$

- Il suffit alors de **calculer** les valeurs de $V(i, i+k)$ pour $k = 0, 1, \ldots, n-1$ et $i = 1, 2, \ldots, n-k$.

Question 3

Stratégie

$i \backslash j$	1	2	3	4	5	6	7	8
1	5	5	8	8	10	12	11	15
2		3	3	5	7	6	10	10
3			3	3	6	7	6	11
4				2	4	3	7	7
5					4	4	5	7
6						1	3	5
7							3	4
8								4

Question 4

```python
# Donnees du probleme
P = [5,3,3,2,4,1,3,4]
n = len(P)

# Cas de base
V = {}
for i in range(n):
    V[i,i] = P[i]
for i in range(n - 1):
    V[i,i+1] = max(P[i], P[i+1])

# Induction
for k in range(2, n):
    for i in range(n - k):
        j = i + k
        V[i,j] = max(min(V[i+2,j], V[i+1,j-1]) + P[i],\
                     min(V[i,j-2], V[i+1,j-1]) + P[j])

# On affiche la table
s = ''
for i in range(n):
    for j in range(n):
        if (i,j) in V:
            s += '%4s' % V[i,j]
        else:
            s += '%4s' % ' '
    s += '\n'
print s
```