

## Concours Blanc - Epreuve d'informatique

**⚠** Consignes

- La calculatrice n'est **pas autorisée**.
- On pourra toujours librement utiliser une fonction demandée à une question précédente même si cette question n'a pas été traitée.
- Veuillez à présenter vos idées et vos réponses partielles même si vous ne trouvez pas la solution complète à une question.
- La clarté et la lisibilité de la rédaction et des programmes sont des éléments de notation.

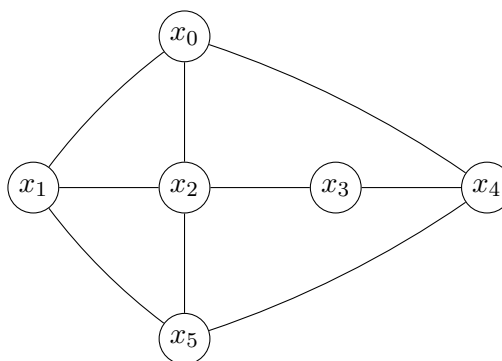
**□ Exercice 1 : Somme maximale de  $k$  termes consécutifs**

On s'intéresse dans cet exercice au calcul de la somme maximale de  $k$  termes consécutifs d'une liste d'entiers. Par exemple, si  $k = 3$  et que la liste est  $[2, 7, -1, 3, 8, -5]$ , la somme maximale de 3 termes consécutifs est 10 (correspondant à la somme  $(-1) + 3 + 8$ ). Dans tout l'exercice, on notera `lst` la liste d'entiers et `n` sa taille, et on supposera que `n > 0` (la liste est non vide) et `k` est inférieur ou égal à `n`.

- Q1–** Ecrire une fonction `sommek_iter` *itérative* qui prend en paramètre une liste `lst`, un entier `k` et un entier `i` et renvoie la somme des `k` termes consécutifs de `lst` à partir de l'indice `i`. C'est à dire que `sommek(lst, k, i)` renvoie `lst[i] + lst[i+1] + ... + lst[i+k-1]`. On supposera que cette somme est définie, c'est à dire que `i ≥ 0` et `i+k-1` est inférieur strictement à la longueur de `lst`.
- Q2–** Ecrire une version *réursive* de la fonction `sommek_iter`.
- Q3–** En utilisant la fonction `sommek` (dans sa version itérative ou réursive), écrire une fonction `sommek_max` qui prend en paramètre une liste `lst` et un entier `k` et renvoie la somme maximale de `k` termes consécutifs de `lst`. On procédera en testant toutes les valeurs possibles de l'indice de départ `i` pour calculer la somme de `k` termes consécutifs.
- Q4–** On veut maintenant exprimer la complexité de `max_sommek` en nombre d'additions. Donner le nombre d'additions lors d'un appel à `sommek` et en déduire en fonction de `k` et `n` le nombre d'additions effectuées par `sommek_max`.
- Q5–** En observant que la somme des `k` premiers termes à partir de l'indice `i+1` s'obtient à partir de celle à l'indice `i` en effectuant seulement deux additions, proposer et écrire une nouvelle version plus efficace de la fonction `max_sommek`.
- Q6–** Quelle est la complexité en nombre d'additions de cette nouvelle version en fonction de `n` et de `k` ?

**□ Exercice 2 : Coloration d'un graphe**

Dans toute la suite de l'exercice, on considère un graphe non orienté  $G = (S, A)$  qu'on supposera représenté en Python par un dictionnaire dont les clés sont les sommets et les valeurs les listes d'adjacence. Par exemple, le graphe  $G$  suivant :



est représenté par le dictionnaire suivant :

```
{'x0': ['x1', 'x2', 'x4'],  
  'x1': ['x0', 'x2', 'x5'],  
  'x2': ['x0', 'x1', 'x3', 'x5'],  
  'x3': ['x2', 'x4'],  
  'x4': ['x0', 'x3', 'x5'],  
  'x5': ['x1', 'x2', 'x4']}
```

Le but de l'exercice est de colorer les sommets du graphe  $G$  de sorte que deux sommets adjacents n'aient pas la même couleur. On représente une coloration par un dictionnaire dont les clés sont les sommets et les valeurs sont des entiers représentant la couleur du sommet. Par exemple, la coloration suivante est valide :