


 La calculatrice est interdite. Écrire en langage Python en faisant attention à bien indenter.

Exercice 1 (piles) :

Lors de l'utilisation des piles, on pourra utiliser les 6 fonctions suivantes :

fonction	creer_pile()	empiler(p,e)	depiler(p)	taille(p)	est_vider(p)	sommet(p)
variables	\emptyset	p : pile e : element	p : pile	p : pile	p : pile	p : pile
sortie	une pile	\emptyset	un élément	un entier	un booléen	un élément

Pour les deux questions suivantes, on considère des piles ne contenant que des nombres flottants.

- Écrire une fonction `ordreCroissant(p)` qui prend en argument une pile `p` et renvoie `True` si la pile est ordonnée dans l'ordre croissant, `False` sinon ( la base devra être le plus petit élément).
- Écrire une fonction `insertion(p, x)` qui prend en argument une pile `p` ordonnée dans l'ordre croissant et un nombre `x`, et qui renvoie une nouvelle pile toujours ordonnée dans laquelle le nombre `x` a été inséré.
- Écrire une fonction `convertStr2piles(phrase)` qui prend en argmuent une chaîne de caractères et qui renvoie une pile composée de chacun des caractères dans l'ordre.
expl : `convertStr2piles("cpge")` renvoie une pile de hauteur 4 dont le sommet est "e")
- Écrire une fonction `extractFromPiles(p, i)` qui prend en argument une pile `p` et un entier non nul `i` et qui renvoie la pile sans le i^e élément en partant du sommet. Si la pile ne possède pas de i^e élément, elle est renvoyée telle quelle.

Exercice 2 (récursivité) :

On définit la fonction suivante

```
def f(n):
    if n<=100: return f(f(n+11))
    else : return n-10
```

- que renvoie l'appel `f(101)` ?
- plus généralement, que renvoie `f(n)` lorsque $n > 100$?
- que renvoie `f(100)` ? `f(99)` ? `f(98)` ?
- Conjecturer ce que renvoie `f(n)` lorsque $n \leq 100$, puis démontrer que votre conjecture est correcte. (on pourra faire 2 cas : $90 \leq n \leq 101$ et $a \leq n \leq a + 11$ (avec $a < 91$)).
- Quel est le coût temporel de `f(n)` ?

Exercice 3 (récursivité) :

On définit un mot comme un demi-palindrome si les lettres de sa première moitié pris une fois sur deux sont égales aux lettres de la seconde moitié prise depuis la fin du mot une fois sur deux. *i.e.* si la 1^{ère} et la dernière lettre ; la 3^e et la 3^e en partant de la fin ; etc. sans jamais dépasser le milieu sont 2 à 2 identiques.

Ainsi "a" ; "abca" ; "abcea" ; "abcdcea" sont des demi-palindrommes mais "ab" ; "abcb" ; "abceb" ; "abcddba" n'en sont pas.

- Écrire une fonction récursive `demipalindrome(m)` qui prend en argument une chaîne de caractères `m` et qui renvoie `True` si `m` est un demi-palindrome, `False` sinon.
- Quel est le coût temporel de `demipalindrome(m)` ?