

DST 1 IPT

Corrigé

Les commentaires sont en rouge.

Exercice 1 - Définir une fonction.

1. Si on ne sait pas écrire cette fonction, il faut s'y mettre sérieusement :

```
def fcarre(x)
    y = x * x # Eviter d'écrire x = x * x même si cela marche ici.
    return y
```

Notez qu'il n'est en fait même pas nécessaire d'introduire y :

```
def fcarre(x)
    return x * x
```

2. Même contexte, mais avec une approche conditionnelle, il faut donc utiliser *if*

```
def fbizarre(n)
    if n%2 == 0 :
        return n/2
    else :
        return (n + 1)/2
```

Exercice 2 - Manipulation des listes.

1. On utilise $len(L)$.
2. Fait en TP :

```
def produit(L, M)
    if len(L) == len(M) :
        P = []
        for i in range(len(L)):
            P.append(L[i] * M[i])
        return(P)
    else :
        return("problème de longueur")
```

Exercice 3 - Manipulation des chaînes.

1. On triche un peu avec la commande « x in L » qui renvoie un booléen selon que x soit dans une liste L ou pas. Sans cette commande, on doit faire une boucle for.

```
presence_a(chaine)
    return ('a' in chaine)
```

2. On parcourt la chaîne de caractère avec une boucle for, en incrémentant un compteur à chaque occurrence de la lettre 'b' :

Algorithme :

```
def nombre_b(chaine)
    n=len(chaine)
    compteur=0 # on crée un compteur : une variable à incrémenter
    for i in range(n):
        if chaine[i]=='b': # on teste si les éléments de la chaînes sont des b
            compteur+=1 # on incrémente le compteur
    return compteur
```

Exercice 4 - Analyse d'algorithme.

1. Les valeurs successives de p sont 5, 20, 60, 120, 0
2. A l'avant dernière étape, p vaut $n!$, puis à la dernière on multiplie par 0. Donc le programme renvoie toujours 0. Une bourde du programmeur ?