

S'entraîner pour le bac

1

Exercice tiré de l'exercice 3 du sujet métropole
du MERCREDI 19 JUIN 2024



La fonction `construction_arbre` fait appel à la fonction `inserer` qui prend pour paramètre `arb`, de type `Noeud`, et `eq`, de type `Equipe`. Cette fonction construit le nœud à partir de `eq` et l'insère à sa place dans l'ABR.

```

1 def inserer(arb, eq):
2     """ Insertion d'une équipe à sa place dans un ABR contenant
3         au moins un noeud. """
4     if convert(eq.temps_etape) < convert(arb.racine.temps_etape):
5         if arb.gauche is None:
6             arb.gauche = ...
7         else:
8             inserer(..., eq)
9     else:
10        if arb.droit is None:
11            arb.droit = Noeud(eq)
12        else:
13            ...

```

10. Expliquer en quoi la fonction `inserer` est une fonction récursive.

2

Exercice tiré de l'exercice 3 du sujet métropole
du JEUDI 20 JUIN 2024



```

1 def parcours(G, depart, chemin = [], lst_chemins = []):
2     if chemin == []:
3         chemin = [depart]
4     for sommet in voisins(G, depart):
5         if sommet not in chemin:
6             lst_chemins.append(chemin + [sommet])
7             parcours(G, sommet, chemin + [sommet])
8     return lst_chemins

```

10. Expliquer en quoi la fonction `parcours` est une fonction récursive.

S'entraîner pour le bac

3

Exercice tiré de l'exercice 3 du sujet de la reunion
De juin 2023



4. La fonction `est_present(n, car)` permet de tester si le caractère `car` est présent ou non dans l'arbre `n` de type `Noeud`.

```

1. def est_present(n, car) :
2.     if n == ..... :
3.         return False
4.     elif n.valeur == ..... :
5.         return True
6.     else :
7.         return est_present(n.droite, car) or .....
```

b. La fonction `est_present` est-elle récursive ? Justifier votre réponse.

4

Exercice tiré de l'exercice 2 du sujet 0
De 2023



- a. Expliquer en quelques mots ce qu'est une fonction récursive.
b. On considère la fonction Python suivante :

Numéro de lignes	Fonction <code>compte_rebours</code>
1	<code>def compte_rebours(n) :</code>
2	<code> """ n est un entier positif ou nul """</code>
3	<code> if n >= 0:</code>
4	<code> print(n)</code>
5	<code> compte_rebours(n - 1)</code>

L'appel `compte_rebours(3)` affiche successivement les nombres 3, 2, 1 et 0. Expliquer pourquoi le programme s'arrête après l'affichage du nombre 0.

5

**Exercice tiré de l'exercice 2 du sujet 0
De 2023**

3. La fonction `somme_entiers_rec` ci-dessous permet de calculer la somme des entiers, de 0 à l'entier naturel `n` passé en paramètre.

Par exemple :

- Pour `n = 0`, la fonction renvoie la valeur 0.
- Pour `n = 1`, la fonction renvoie la valeur $0 + 1 = 1$.
- ...
- Pour `n = 4`, la fonction renvoie la valeur $0 + 1 + 2 + 3 + 4 = 10$.

Numéro de lignes	Fonction <code>somme_entiers_rec</code>
1	<code>def somme_entiers_rec(n):</code>
2	<code> """ Permet de calculer la somme des entiers,</code>
3	<code> de 0 à l'entier naturel n """</code>
4	<code> if n == 0:</code>
5	<code> return 0</code>
6	<code> else:</code>
7	<code> print(n) #pour vérification</code>
8	<code> return n + somme_entiers_rec(n - 1)</code>

L'instruction `print(n)` de la ligne 7 dans le code précédent a été insérée afin de mettre en évidence le mécanisme en œuvre au niveau des appels récursifs.

a. Écrire ce qui sera affiché dans la console après l'exécution de la ligne suivante :

```
res = somme_entiers_rec(3)
```

b. Quelle valeur sera alors affectée à la variable `res` ?