

TP1_SecondeELEVE

June 8, 2020

1 TP1 : Deux grands théorèmes de géométrie à la sauce Python

1.1 Prise en main du langage Python

1. Quelques calculs pour prendre en main le langage Python

- Addition de deux valeurs entières
- Soustraction de deux valeurs entières (à compléter)
- Multiplication de deux valeurs entières (à programmer)
- Division de deux valeurs entières (à programmer)

```
[4]: from math import *      # importation nécessaire pour disposer de la bibliothèque
      ↪ mathématique

def addition(a,b):         #fonction permettant de faire l'addition de deux valeurs
    return a+b

def soustraction(a,b):
    return ....

# Programmer la multiplication entre deux valeurs

# Programmer la division entre deux valeurs

#Partie nécessaire pour exécuter un programme
print(addition(4,6))
```

10

2. Quelques fonctions mathématiques nécessaires

- Le calcul de puissance d'un nombre entier
- Le calcul de la racine carré d'un nombre entier (faire une recherche de la fonction Python utile)

```
[3]: from math import *
```

```

def puissance2(n):
    return n**2

# programmer la fonction racinecarre prenant en argument (paramètre) le nombre
↳ entier n

#Partie nécessaire à l'exécution des programmes
print(puissance2(11))

```

121

1.2 Théorème de Pythagore

1. Créer un programme Python renvoyant la longueur du troisième côté connaissant les deux autres pour un triangle rectangle. Exécuter-le avec comme longueurs connues : 3 et 4.

```

[11]: from math import *

#Programme Théorème de Pythagore
def ThPythagore(a,b):
    .....          #à compléter

#Partie nécessaire à l'exécution du programme
print(.....) #à compléter

```

5.0

2. Exécuter-le avec les longueurs 6 et 8. **Réponse :**

1.3 Réciproque/Contraposée du Théorème de Pythagore

1. Créer un programme Python renvoyant Vrai si le triangle est rectangle et faux sinon connaissant la longueur des trois côtés d'un triangle. Exécuter-le avec comme longueurs connues le triplet (3,4,5).

```

[13]: from math import *

#Programme Réciproque/Contraposée du Théorème de Pythagore
def RecContThPythagore(a,b,c):
    if (a>b and a>c):
        if .....: #à compléter
            return True
        else:
            return False
    elif (.....): #à compléter
        if b**2==a**2+c**2:
            return True

```

```

else:
    return False
else:
    if .....: #à compléter
        return True
    else:
        return False

#Partie nécessaire à l'exécution du programme
print (.....) #à compléter

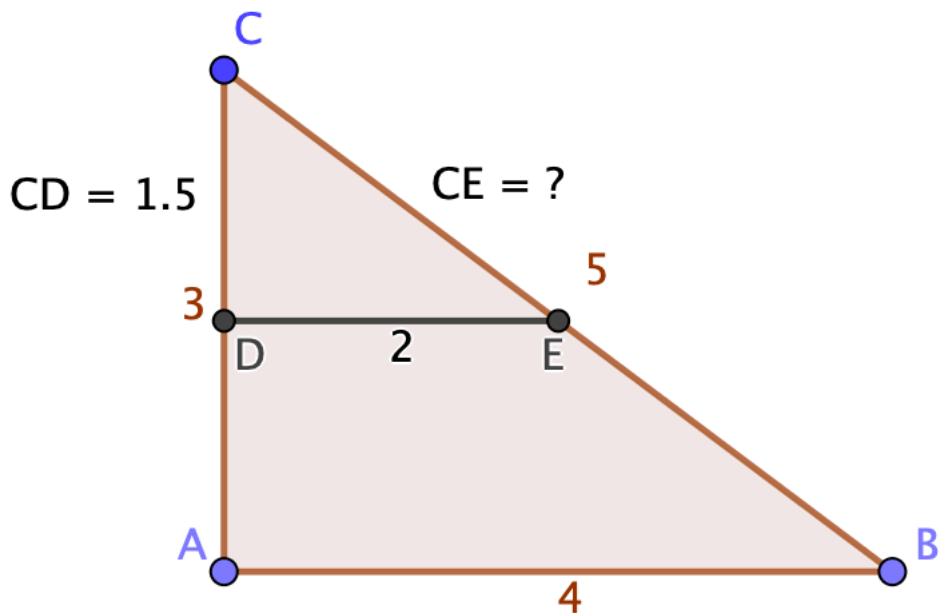
```

False
True
True

2. Exécuter-le avec le triplet (10,6,8). **Réponse :**

1.4 Théorème de Thalès

1. Créer un programme Python renvoyant la longueur recherchée connaissant les trois autres longueurs dans la configuration donnée ci-après en admettant que les arguments a,b,c et d représentent respectivement les longueurs des côtés [AE], [AB] et [CD]. Exécuter-le avec les données présentes sur la configuration.



```

[19]: from math import *

# Programme du théorème de Thalès (a/b = c/d)

```

```
#Partie nécessaire à l'exécution du programme  
print(.....)
```

2.5

1.5 Et si on testait tout cela avec d'autres valeurs...

1. Tester les deux programmes portant sur le Théorème de Pythagore avec:
 - (3.2,4.7) avec le théorème de Pythagore.
 - (3.2,4.7, ?) la valeur manquante étant celle trouvée dans le point précédent.
 - Comment expliquer la non-cohérence entre les deux résultats ?
2. Tester le programme de la Réciproque/Contraposée du Théorème de Pythagore avec:
 - (2.4,3.2,4) Que se passe-t-il ?
 - (1/3,(sqrt(8))/3,1) Que se passe-t-il ?
 - (3/7,4/7,5/7) Que se passe-t-il ?
 - Expliquer les raisons d'une éventuelle incohérence.