

Python

Exercise 1

- Ecrire un script permettant de rentrer 1 caractères et un entier
- Afficher la variables sous forme d'entiers et de caractères

L'affichage devra ressembler aux lignes ci-dessous :

```
rentrer un caractère : A
rentrer un entier    : 66
A vaut 65
66 vaut B
Voulez-vous recommencer [oO] ? N
Merci a bientôt
```

TP 1.2 – Calcul d'une surface

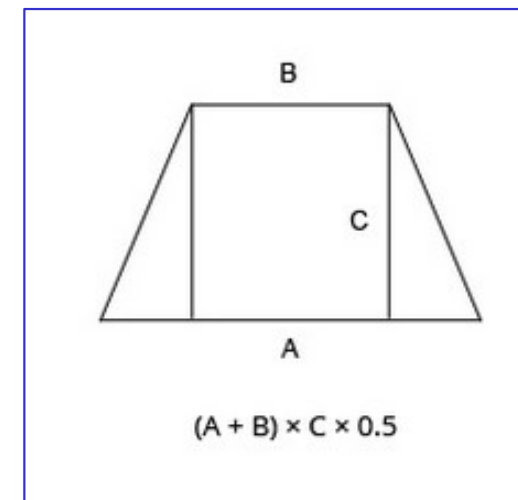
- **Ecrire un programme qui calcule la surface d'un trapèze**
- L'affichage devra ressembler aux lignes ci-dessous :

Rentrer A (en m) : 2.

Rentrer B (en m) : 5.

Rentrer C (en m) :10.

La surface du trapèze est de 35 m



Challenge : dessiner avec python le trapèze

Ecrire un algorithme qui demande un nombre entier positif de départ, et qui calcule la somme et le factoriel de l'entiers

- Exemple d'exécution demandé :

```
entrer un entier positif : 5
```

```
1 + 2 + 3 + 4 + 5 = 15
```

```
15 = 1 + 2 + 3 + 4 + 5
```

```
1 * 2 * 3 * 4 * 5 = 120
```

```
120 = 1 * 2 * 3 * 4 * 5
```

```
Voulez-vous recommencer [oO] ? N
```

```
Merci a bientôt
```

Afficher un arbre de Noël en utilisant les symboles = et *. La hauteur de l'arbre sera donnée par l'utilisateur, et l'affichage devra ressembler aux lignes ci-dessous :

Hauteur de l'arbre : 5

```
=====*=====
=====* * *=====
=====* * * * *=====
=====* * * * * * *=====
=====* * * * * * * * *=====
=====*=====
=====* * *=====
```

- Ecrire un script permettant de saisir un entier au clavier et d'afficher :
 - Le logarithme népérien de l'entier
 - Le sinus de l'entier
 - Le cosinus de l'entier

Voir la bibliothèque python math

Soit la fonction ***factoriel(n) = 1 * 2 * 3 * 4 * n-1 * n (= n!)*** avec n entier n > 0

Soit la fonction ***f1(x,n) = xⁿ / factoriel(n)*** avec n entier n > 0 et x réel

Ecrire les fonctions **factoriel** et **f1** qui calcule **f1(x,n)** avec n entier et x réel

Soit N un entier rentrer au clavier

Soit X un réel inferieur a 0 rentrer au clavier

Créer une fonction **Res(X,N)** qui calculer somme tel que :

$$\text{Res} = f1(X,1) + f1(X,2) + + f1(X,N-1) + f1(X,N)$$

$$\text{Res} = \sum_{i=1}^N f1(X, i) = 1 + x / 1! + x^2 / 2! + + x^N / (N-1)! + x^N / N!$$

Tester les fonctions: ***factoriel(n)***, ***f1(x,n)*** , ***Res(X,N)***

Comparer le résultats de la fonction Res avec ***math.exp(x)***

Challenge : Optimiser la fonction Res(X,N)

TP 1.7 : Fonctions

Soit la suite U tel que :

$$U_n = U_{n-1} + 1 / n! \quad \text{avec } n > 0 \text{ et } U_0 = 1$$

Ecrire la fonction U(n) qui rends la valeur de U avec n un entier le résultat de U est un réel

Soit la suite V tel que :

$$V_n = U_n + 1 / (n * n!)$$

Ecrire la fonction V(n) qui rends la valeur de V avec n un entier le résultat de V est un réel

Tester les fonctions U(n) et V(n)

Challenge : Optimiser les fonctions U(n) et V(n)

TP 1.8 : le Tiercé

Écrire un algorithme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires.

On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

- Dans l'ordre : une chance sur X de gagner
- Dans le désordre : une chance sur Y de gagner

X et Y nous sont donnés par la formule suivante, si n est le nombre de chevaux partants et p le nombre de chevaux joués (on rappelle que le signe ! signifie "factorielle", comme dans l'exercice 1.8) :

- $X = n ! / (n - p) !$
- $Y = n ! / (p ! * (n - p) !)$

NB :

cet algorithme peut être écrit d'une manière simple, mais relativement peu performante. Ses performances peuvent être singulièrement augmentées par une petite astuce.

Vous commencerez par écrire la manière la plus simple, puis vous identifierez le problème, et écrirez une deuxième version permettant de le résoudre.