

DIFFÉRENTES PRÉSENTATIONS D'ALGO

Variables :
 x_A est de type

 y_A est

 x_B est

 y_B est

 x_I est

 y_I est

Initialisation, entrées :

Saisir

Saisir

Saisir

Saisir

Traitement :

.....

.....

Sortie :

Afficher

Afficher

Voici les algorithmes version “calculatrice” pour la question ② :

Nouvelle commande !! \casiotriangle

ALGO**ALGO****Variables :**

x, y sont des réels

Initialisation, entrées :

$-1 \rightarrow x$

$x^3 - 5x \rightarrow y$

Traitement :

Tant que $y > 3$ Faire

..... $\rightarrow x$

..... $\rightarrow y$

Fin Tant que

Sortie :

Afficher $(x - 0,01 ; x)$

Avec une TI :

:Input "Nombre à dépasser",A

:3000 \rightarrow S

:0 \rightarrow N

:While $U \leq A$

:1,04S+350 \rightarrow S

:N+1 \rightarrow N

:End

:Disp "On dépassera la somme au bout de

:",N,"années"

Avec une Casio :

"Nombre à dépasser"↵

? \rightarrow A↵

3000 \rightarrow S↵

0 \rightarrow N↵

While $S \leq A$ ↵

1,04S+350 \rightarrow S↵

N+1 \rightarrow N↵

WhileEnd↵

"On dépassera la somme au bout de : " ▲

N ▲

"années"↵

Variables :

1. J et N sont entiers naturels
2. p est un nombre réel

Entrée :

3. Saisir N

Initialisation :

4. p prend la valeur 1

Traitement :

5. Pour J allant de 1 à N
6. p prend la valeur
7. Fin Pour

Sortie :

8. Afficher p

ALGO**Variables :**

x, y sont des réels

Initialisation, entrées :

$-1 \rightarrow x$

$x^3 - 5x \rightarrow y$

Traitement :

Tant que $y > 3$ Faire

..... $\rightarrow x$

..... $\rightarrow y$

Fin Tant que

Sortie :

Afficher $(x - 0,01; x)$

ALGO

```
# Calcul de la factorielle
def factorielle(x):
    if x < 2:
        return 1
    else:
        return x * factorielle(x-1)
str(5) + "! = " + str(factorielle(5))
```

ALGO

```
# Calcul de la factorielle
def factorielle(x):
    if x<2:
        return 1
    else:
        return x*factorielle(x-1)
str(5)+"! = "+str(factorielle(5))
```

Lorsque le code est à insérer dans un autre environnement (par exemple TikZ), cela peut poser problème.

Voici un exemple de solution :

Dans le préambule, on écrit :

`\newsavebox{\codeone} % (le nom codeone est à choisir)`

On stocke le code dans une `lrbox` (déclaration globale utilisable dans tout le document) en dehors de l'autre environnement mais entre `\begin{document}` et `\end{document}`.

```
\begin{lrbox}{\codeone}
\begin{lstlisting}[language=Python]
print("Hello World!")
\end{lstlisting}
\end{lrbox}
```

Puis on insère le code suivant à l'endroit voulu : `\usebox{\codeone}`

Et on obtient :

```
print("Hello World!")
```

ALGO

Avec `algorithm2e`

Algorithme MaxCompatible(S)

```
1  $S' \leftarrow \emptyset$ 
2 Trier les activités de S par durée croissante
3 pour  $i$  de 1 à  $|S|$  faire
4   si l'activité  $i$  est compatible avec les activités de  $S'$  alors
5     Ajouter l'activité  $i$  à  $S'$ 
```

Algorithme de dichotomie

```
1 Entrer  $a, b, p$ 
2 tant que  $b - a > p$  faire
3    $\frac{a + b}{2} \rightarrow c$ 
4   si  $f(c) \times f(a) > 0$  alors
5      $c \rightarrow a$ 
6   sinon
7      $c \rightarrow b$ 
```