BCPST 1 INFORMATIQUE

FONCTIONS, TESTS, INSTRUCTIONS CONDITIONNELLES, BOUCLES

Exercices du programme de colle :

1. Définir en langage *Python* la fonction : $x \longmapsto \ln(3+\cos^2 x)$.
```
from numpy import log, cos
def f(x) :
    return log(3+cos(x)**2)
```
2. Définir en langage *Python* la fonction : $(x,y) \longmapsto \sin^3(2x)+\cos(xy)$.
```
from numpy import sin, cos
def f(x,y) :
    return sin(2*x)**3 + cos(x*y)
```
3. Écrire une fonction d'argument n renvoyant : $\displaystyle\sum_{k=1}^{n} \ln(k+1)\cos(k)$.
```
from numpy import log, cos
def f(n) :
    S = 0
    for k in range(1,n+1) :
        S = S + log(k+1)*cos(k)
    return S
```
4. Définir la fonction factorielle.
```
def fact(n) :
    P = 1
    for k in range(1,n+1) :
        P = P*k
    return P
```
5. Écrire une fonction d'argument x renvoyant : $\begin{cases} 2x+1 & \text{si } x>0 \\ \cos(x) & \text{sinon} \end{cases}$
```
from numpy import cos
def f(x) :
    if x > 0 :
        return 2*x+1
    else :
        return cos(x)
```
6. Définir la fonction `suite(n)` renvoyant u_n lorsque $\begin{cases} u_0 = 5 \\ \forall n \geqslant 0,\ u_{n+1} = \dfrac{1+u_n}{1+u_n^2} \end{cases}$
```
def suite(n) :
    u = 5
    for k in range(n) :
        u = (1+u) / (1+u**2)
    return u
```

Exercices d'entraînement :

1. Définir une fonction d'argument $n \in \mathbf{N}$ renvoyant 1 si n est pair et non multiple de 3, et 0 sinon.
   `n%2 == 0` permet de tester si n est pair,
   `n%3 == 0` teste si n est multiple de 3.
2. Définir une fonction `somme` d'arguments $n, p \in \mathbf{N}^\star$ renvoyant : $1+2^n+3^n+\ldots+p^n$.
3. Définir une fonction `sommeLn` d'arguments $n \in \mathbf{N}^\star$ renvoyant : $\ln 1+\ln 2+\ln 3+\ldots+\ln n$.

4. Définir une fonction `produit` d'arguments $n \in \mathbf{N}^\star, x \in \mathbf{R}$ renvoyant :
$$\sin(x) \times \sin(2x) \times \sin(3x) \times \ldots \times \sin(nx).$$
5. Définir une fonction d'argument $n \in \mathbf{N}^\star$, renvoyant $\displaystyle\sum_{k=1}^{n} \frac{1}{k\sqrt{k}}$.
6. Définir une fonction d'argument $n \in \mathbf{N}^\star$, renvoyant $\displaystyle\sum_{k=1}^{n} \frac{\sqrt{k}}{1+k^2}$.
7. Définir une fonction d'arguments $n \in \mathbf{N}, A \in \mathbf{R}$ et renvoyant le booléen `True` si $\displaystyle\sum_{k=1}^{n} \frac{1}{k} \geqslant A$, et `False` sinon.
8. Définir une fonction d'argument $n \in \mathbf{N}^\star$ renvoyant $\displaystyle\prod_{k=2}^{n} u_k$, où $u_k = 1 + \frac{1}{k}$ si k est multiple de 3, $u_k = 1 - \frac{1}{k}$ si k est multiple de 4 et non multiple de 3, et $u_k = 1 - \frac{1}{k^2}$ sinon.
9. Définir une fonction d'arguments n entier et x flottant, renvoyant : $\displaystyle\sum_{k=1}^{n} \frac{kx}{1+kx^2}$.
10. Définir une fonction d'arguments n entier et x flottant, renvoyant : $\displaystyle\prod_{k=1}^{n} \frac{\sin(kx)}{1+k^2}$.
11. Définir une fonction d'arguments n entier et x flottant, renvoyant : $\displaystyle\sum_{k=2}^{n} \frac{x^k}{k\ln(k)}$.
12. Définir la fonction $f_n : x \longmapsto 1 + x\cos(x) + x^2\cos(2x) + x^3\cos(3x) + \cdots + x^n\cos(nx)$.
13. Définir une fonction d'arguments $n \in \mathbf{N}, x \in \mathbf{R}$ renvoyant : $\displaystyle\prod_{\substack{k=1 \\ \sin(k)>x}}^{n} \sin(k)$.
14. Définir une fonction d'arguments $n \in \mathbf{N}, x \in \mathbf{R}$ renvoyant : $\displaystyle\sum_{\substack{k\geqslant 1 \\ k\ln(k)<x}} \frac{1}{k}$.
15. Définir une fonction `u(n)` renvoyant u_n lorsque $\begin{cases} u_0 = 1 \\ \forall n \geqslant 0,\ u_{n+1} = \sin(u_n) \end{cases}$
16. Définir une fonction `v(n)` renvoyant v_n lorsque $\begin{cases} v_0 = 0 \\ \forall n \geqslant 0,\ v_{n+1} = \sqrt{1+v_n} \end{cases}$
17. Définir une fonction `w(n)` renvoyant w_n lorsque $\begin{cases} w_0 = w_1 = 1 \\ \forall n \geqslant 0,\ w_{n+2} = 2w_{n+1} - 3w_n \end{cases}$
18. Pour $n \in \mathbf{N}^\star$, on pose $u_n = n + \sqrt{n} + 2\ln(n)$.
Écrire une fonction d'argument M renvoyant le plus petit indice n tel que $u_n \geqslant M$.
19. Pour $n \in \mathbf{N}^\star$, on pose $u_n = \dfrac{2+\sin(n)}{n}$.
Écrire une fonction d'argument $\varepsilon > 0$ renvoyant le plus petit indice n tel que $u_n < \varepsilon$.
20. Écrire une fonction sans argument, renvoyant le plus petit entier naturel n pouvant s'écrire de 2 façons différentes comme somme de 2 cubes.
$n = \min\left\{n \in \mathbf{N},\ \exists (a,b,c,d) \in \mathbf{N}^4,\ n = a^3 + b^3 = c^3 + d^3 \ \wedge\ a \neq c \ \wedge\ a \neq d\right\}$
(on admet que cet ensemble est non vide)
Comment modifier cette fonction pour qu'elle renvoie les N plus petits entiers n possédant cette propriété ?