BCPST 2$_A$

Correction du devoir surveillé du samedi 6 septembre 2025

EXERCICE 1

1)
```
def u(n):
    a, b = -2, -2
    for k in range(n):
        a, b = b, 4*b - 4*a          # A la fin de la boucle a = u(k+1) et b = u(k+2)
    return a
```
Remarque : la fonction précédente donne le bon résultat pour $n = 0$ et $n = 1$.

2) On reconnait une suite récurrente linéaire d'ordre 2 à coefficients constants d'équation caractéristique :

$$x^2 - 4x + 4 = 0 \qquad (\iff (x-2)^2 = 0 \ )$$

Cette équation a une unique racine réelle : 2 donc il existe α et β tels que : $\forall n \in \mathbb{N}, \quad u_n = (\alpha + \beta n)2^n$.
de plus α et β vérifient le système :

$$\begin{cases} \alpha \quad = \ -2 \ (= u_0) \\ 2\alpha + 2\beta = -2 \ (= u_1) \end{cases} \iff \begin{cases} \alpha = -2 \\ \beta = 1 \end{cases}$$

En conclusion :

$$\boxed{\forall n \in \mathbb{N}, \quad u_n = (n-2)\, 2^n}$$

3) a)
```
def C(n):
    S = 0
    for k in range(n+1):
        S += math.comb(n, k)*u(k)
    return S
```
b)
```
def C2(n):
    a, b = -2, -2
    c =  n
    S = a + c*b
    for k in range(1,n):
        c = (n-k)*c//(k+1)              # c contient (k+1) parmi n
        a, b = b, 4*b-4*a               # b contient u(k+1)
        S += c*b
    return S
```
c) $\displaystyle\sum_{k=0}^{n} \binom{n}{k} 2^k = (1+2)^n$ *(formule du binôme)*

$$\boxed{\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n}$$

$$\begin{aligned} \sum_{k=0}^{n} k\binom{n}{k} 2^k &= \sum_{k=1}^{n} k\binom{n}{k} 2^k && \textit{(pour } k=0,\ k\tbinom{n}{k}2^k = 0 \textit{)} \\ &= \sum_{k=1}^{n} n\binom{n-1}{k-1} 2^k && \textit{(formule du chef)} \\ &= 2n\sum_{k=1}^{n} \binom{n-1}{k-1} 2^{k-1} \\ &= 2n\sum_{k=0}^{n-1} \binom{n-1}{k} 2^k && \textit{(decalage d'indice)} \end{aligned}$$

et avec la formule du binôme on obtient :

$$\boxed{\sum_{k=0}^{n} k\binom{n}{k} 2^k = 2\, n\, 3^{n-1}}$$

d)

$$\begin{aligned}
\sum_{k=0}^{n} \binom{n}{k} u_k &= \sum_{k=0}^{n} \binom{n}{k} (k-2)\, 2^k \qquad \textit{(avec le résultat de la question 2))} \\
&= \sum_{k=0}^{n} k \binom{n}{k} 2^k - 2 \sum_{k=0}^{n} \binom{n}{k} 2^k \\
&= 2n3^{n-1} - 2\, 3^n \qquad \textit{(avec les résultat de la question 3)c)}
\end{aligned}$$

$$\boxed{\sum_{k=0}^{n} \binom{n}{k} u_k = 2\,(n-3)\; 3^{n-1}}$$

e) C_n est le produit de deux suites tendant vers $+\infty$ donc $\boxed{(C_n) \text{ diverge vers } +\infty}$

f) On utilise la formule de C_n trouvé en 3)d)

```
n = 1
while 2*(n-3)*3**(n-1) <= 100000:
    n += 1
print(n)
```

4) a) C'est une simple somme télescopique :

$$\sum_{k=0}^{n-2} (4\, u_{k+1} - 4\, u_k) = 4u_{n-1} - 4u_0$$

et avec le résultat de la question 2) :

$$\boxed{\sum_{k=0}^{n-2} (4\, u_{k+1} - 4\, u_k) = (n-3)\, 2^{n+1} + 8}$$

b) Comme pour tout entier n, $\quad u_{n+2} = 4\, u_{n+1} - 4\, u_n \quad$ on a :

$$\sum_{k=0}^{n-2} (4\, u_{k+1} - 4\, u_k) = S_n - u_0 - u_1$$

$$\boxed{S_n = (n-3)\, 2^{n+1} + 4}$$

EXERCICE 2

1)
```
def pyramide(n):
    L = [ [1] ]
    for k in range(2, n+1):
        A = [L[-1][-1] + 2]          # la liste A va contenir la k ième ligne de la pyramide
        for j in range(k-1):
            A.append(A[-1] + 2)      # on ajoute k-1 nombres impairs
        L.append(A)
    return L
```

2) A la k ième ligne il y a k nombres donc sur les n premières lignes il y a $\sum_{k=1}^{n} k$ nombres.

$$\boxed{N_n = \frac{n(n+1)}{2}}$$

3) Notons $(u_n)_{n\geqslant 1}$ la suite des termes impairs, $u_1 = 1$ et $\forall n \in \mathbb{N}, \quad u_n = 1 + 2 \times (n-1)$

D_n est égale à u_{N_n}, (c'est le N_n-ième nombre impair) donc $D_n = 1 + 2 \times \left(\frac{n(n+1)}{2} - 1\right)$

$$\boxed{D_n = n^2 + n - 1}$$

4) pour $n \geqslant 2$, G_n est le nombre impair juste après D_{n-1} donc $G_n = D_{n-1} + 2$
ou encore $G_n = (n-1)^2 + (n-1) - 1 + 2 = n^2 - n + 1$ (expression aussi vraie pour $n = 1$)

$$\boxed{G_n = n^2 - n + 1}$$

.

5) S_n est la somme des premiers termes d'une suite arithmétique donc :

$$S_n = N_n \times \frac{1 + D_n}{2}$$

En utilisant les résultats précédents il vient :

$$\boxed{S_n = \frac{n^2(n+1)^2}{4}}$$

6) ℓ_n est la somme de n termes consécutifs d'une suite arithmétique donc :

$$\ell_n = n\frac{G_n + D_n}{2}$$

il vient en utilisant les expressions précédentes :

$$\boxed{\ell_n = n^3}$$

7) Avec le même raisonnement on montre que pour chaque $k \in [\![1, n]\!]$, $\ell_k = k^3$
et comme $S_n = \displaystyle\sum_{k=1}^{n} \ell_k$ on a $S_n = \displaystyle\sum_{k=1}^{n} k^3$
or on a vu à la question 4) que $S_n = \dfrac{n^2(n+1)^2}{4}$, donc on a bien :

$$\boxed{\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}}$$

EXERCICE 3

1) Montrons par récurrence sur n que pour tout entier naturel $n \geqslant 2$, $\underbrace{u_n > 1}_{\mathcal{P}(n)}$.

- Pour $n = 2$,
  on a $u_0 \geqslant 0$ et $u_1 = \sqrt{u_0} + \frac{1}{0+1}$ donc $u_1 \geqslant 1$,
  or $u_2 = \sqrt{u_1} + \frac{1}{1+1}$ donc $u_2 > 1$
  ce qui donne

$$\mathcal{P}(2) \quad \text{(est vraie)}$$

- Soit n un entier supérieur ou égal à 2 tel que $\mathcal{P}(n)$ est vraie,

  on a $u_n > 1$ et $u_{n+1} = \sqrt{u_n} + \dfrac{1}{n+1}$ donc $u_{n+1} > 1$ ($\mathcal{P}(n+1)$ est vraie),

En conclusion :

$$\boxed{\text{pour tout entier naturel } n \geqslant 2, \quad u_n > 1}$$

2) On suppose que $(u_n)_{n\in\mathbb{N}}$ converge vers un réel ℓ,
on a alors (u_{n+1}) converge vers un réel ℓ et $(\sqrt{u_n})$ converge vers un réel $\sqrt{\ell}$,
et comme $\left[\forall n \in \mathbb{N},\ u_{n+1} = \sqrt{u_n} + \frac{1}{n+1}\right]$ et $\displaystyle\lim_{n\to+\infty}\left(\frac{1}{n+1}\right) = 0$
on obtient :

$$\sqrt{\ell} = \ell$$

on en déduit : $\ell = \ell^2$ et ainsi $\ell = 0$ ou $\ell = 1$
de plus on sait que $[\ \forall n \geqslant 2,\ u_n > 1\ ]$ donc on ne peut pas avoir $\ell = 0$
en conclusion :

$$\boxed{\text{Si } (u_n) \text{ converge vers un réel } \ell, \text{ alors } \ell = 1}$$

3) On suppose dans cette question que $(u_n)_{n\in\mathbb{N}}$ est croissante.

a) La seule limite finie possible de (u_n) est 1,
or (u_n) est croissante et $\forall n \geqslant 2,\ u_n > 1$ donc (u_n) ne peut pas converger vers 1.
la suite (u_n) est donc divergente, et comme (u_n) est croissante :

$$\boxed{(u_n) \text{ diverge vers } +\infty}$$

b) Soit $n \in \mathbb{N}$,

$$\begin{aligned} u_{n+1} - u_n &= \sqrt{u_n} + \frac{1}{n+1} - u_n \\ &= \sqrt{u_n} - \left(\sqrt{u_n}\right)^2 + \frac{1}{n+1} \end{aligned}$$

$$\boxed{\forall n \in \mathbb{N}, \quad u_{n+1} - u_n = \sqrt{u_n}(1-\sqrt{u_n}) + \tfrac{1}{n+1}.}$$

c) On a montré que (u_n) diverge vers $+\infty$ donc $\left(\sqrt{u_n}(1-\sqrt{u_n})\right)$ diverge vers $-\infty$,
et comme $\forall n \in \mathbb{N}, \quad u_{n+1} - u_n = \sqrt{u_n}(1-\sqrt{u_n}) + \frac{1}{n+1}$ et $\displaystyle\lim_{n\to+\infty}\left(\frac{1}{n+1}\right) = 0$ on obtient :

$$\boxed{\lim_{n\to+\infty}(u_{n+1}-u_n) = -\infty}$$

d) $\displaystyle\lim_{n\to+\infty}(u_{n+1}-u_n) = -\infty$ impose que $(u_{n+1}-u_n)$ est à valeurs négatives à partir d'un certain rang,
ce qui contredit que la suite (u_n) soit croissante.
En conclusion :

$$\boxed{\text{La suite } (u_n) \text{ n'est pas croissante.}}$$

4) a) La définition de (u_n) est croissante est : [$\forall n \in \mathbb{N}, \quad u_n \leqslant u_{n+1}$].
Mais ici la suite n'est pas croissante donc

$$\boxed{\text{il existe un entier naturel } n_0 \text{ tel que : } u_{n_0+1} < u_{n_0}}$$

b) Soit $n \in \mathbb{N}^*$,

$$\begin{aligned} u_{n+1} - u_n &= \sqrt{u_n} + \frac{1}{n+1} - \sqrt{u_{n-1}} - \frac{1}{n} \\ &= \sqrt{u_n} - \sqrt{u_{n-1}} + \frac{1}{n+1} - \frac{1}{n} \\ &= \frac{\left(\sqrt{u_n}-\sqrt{u_{n-1}}\right)\left(\sqrt{u_n}+\sqrt{u_{n-1}}\right)}{\sqrt{u_n}+\sqrt{u_{n-1}}} + \frac{n}{n(n+1)} - \frac{n+1}{n(n+1)} \end{aligned}$$

$$\boxed{\forall n \in \mathbb{N}^*, \quad u_{n+1} - u_n = \tfrac{u_n - u_{n-1}}{\sqrt{u_n}+\sqrt{u_{n-1}}} - \tfrac{1}{n(n+1)}.}$$

c) On prend ici un n_0 défini à la question 4) a).
Montrons par récurrence sur n que pour tout entier naturel $n \geqslant n_0$, $\underbrace{u_{n+1} \leqslant u_n}_{\mathcal{P}(n)}$.

❶ Pour $n = n_0$,
la définition de l'entier n_0 donne $u_{n_0+1} \leqslant u_{n_0}$
ce qui donne

$$\mathcal{P}(n_0) \quad \text{(est vraie)}$$

❷ Soit n un entier supérieur ou égal à n_0 tel que $\mathcal{P}(n)$ est vraie,
on a $u_{n+1} - u_n \leqslant 0$ et $u_{n+2} - u_{n+1} = \dfrac{u_{n+1}-u_n}{\sqrt{u_{n+1}}+\sqrt{u_n}} - \dfrac{1}{(n+1)(n+2)}$
donc $u_{n+2} - u_{n+1} \leqslant 0$, et ainsi $\mathcal{P}(n+1)$ est vraie.

En conclusion (de ❶ et ❷) : $\forall n \geqslant n_0,\ u_{n+1} \leqslant u_n$

$$\boxed{\text{la suite } (u_n)_{n\in\mathbb{N}} \text{ est décroissante à partir du rang } n_0}$$

d) La suite à valeurs réelles $(u_n)_{n\in\mathbb{N}}$ est minorée par 1 et décroissante à partir d'un certain rang
donc *(Th. de limite monotone)* elle est convergente,
or on a montré à la question 2) que 1 est la seule limite finie possible de (u_n)

$$\boxed{\text{la suite } (u_n)_{n\in\mathbb{N}} \text{ converge vers 1.}}$$

EXERCICE 4

1)

$$S_0(n) = n \qquad S_1(n) = \frac{n(n+1)}{2} \qquad S_2(n) = \frac{n(n+1)(2n+1)}{6} \qquad S_3(n) = \frac{n^2(n+1)^2}{4}$$

2)

```
def S(p, n):
    T = 0
    for k in range(1, n+1):
        T += k**p
    return T
```

3) a)

$$\begin{aligned} \sum_{k=1}^{n}(k+1)^{p+1} &= \sum_{k=2}^{n+1} k^{p+1} && \textit{(décalage d'indice)} \\ &= \sum_{k=1}^{n} k^{p+1} + \underbrace{(n+1)^{p+1}}_{k=n+1} - \underbrace{1}_{k=1} \end{aligned}$$

on a bien :

$$\boxed{\sum_{k=1}^{n}(k+1)^{p+1} = S_{p+1}(n) + (n+1)^{p+1} - 1}$$

b) *C'est une question de cours.*

$$\boxed{(k+1)^{p+1} = \sum_{\ell=0}^{p+1} \binom{p+1}{\ell} k^{\ell}}$$

c) En sommant l'égalité précédente pour k compris entre 1 et un entier n on obtient :

$$\begin{aligned} \sum_{k=1}^{n}(k+1)^{p+1} &= \sum_{k=1}^{n}\sum_{\ell=0}^{p+1} \binom{p+1}{\ell} k^{\ell} \\ &= \sum_{\ell=0}^{p+1}\sum_{k=1}^{n} \binom{p+1}{\ell} k^{\ell} && \textit{(interversion d'ordre)} \\ &= \sum_{\ell=0}^{p+1} \binom{p+1}{\ell} \sum_{k=1}^{n} k^{\ell} \\ &= \binom{p+1}{p+1} S_{p+1}(n) + \sum_{\ell=0}^{p} \binom{p+1}{\ell} S_{\ell}(n) \end{aligned}$$

on a bien :

$$\boxed{\sum_{k=1}^{n}(k+1)^{p+1} = S_{p+1}(n) + \sum_{\ell=0}^{p} \binom{p+1}{\ell} S_{\ell}(n)}$$

d) Les résultats des questions 3)a) et 3)c) donne l'égalité :

$$(n+1)^p - 1 = \sum_{\ell=0}^{p} \binom{p+1}{\ell} S_{\ell}(n)$$

ou encore :

$$(n+1)^p - 1 = (p+1)S_p(n) + \sum_{\ell=0}^{p-1} \binom{p+1}{\ell} S_{\ell}(n)$$

En isolant $S_p(n)$, on obtient bien :

$$\boxed{S_p(n) = \frac{1}{p+1}\left((n+1)^{p+1} - 1 - \sum_{\ell=0}^{p-1} \binom{p+1}{\ell} S_{\ell}(n)\right)}$$

4) D'après la question précédente avec $p = 4$:

$$\begin{aligned}
S_4(n) &= \frac{1}{5}\left((n+1)^5 - 1 - \sum_{\ell=0}^{3}\binom{5}{\ell}S_\ell(n)\right)\\
&= \frac{1}{5}\left((n+1)^5 - 1 - n - 5\,\frac{n(n+1)}{2} - 10\,\frac{n(n+1)(2n+1)}{6} - 10\frac{n^2(n+1)^2}{4}\right)\\
&= \frac{n+1}{5}\left((n+1)^4 - 1 - 5\,\frac{n}{2} - 5\,\frac{n(2n+1)}{3} - 5\frac{n^2(n+1)}{2}\right)\\
&= \frac{n+1}{5}\left(n^4 + 4n^3 + 6n^2 + 4n - 5\,\frac{n}{2} - 5\,\frac{n(2n+1)}{3} - 5\frac{n^2(n+1)}{2}\right)\\
&= \frac{n(n+1)}{5}\left(n^3 + 4n^2 + 6n + 4 - \frac{5}{2} - 5\,\frac{(2n+1)}{3} - 5\frac{n(n+1)}{2}\right)\\
&= \frac{n(n+1)}{30}\left(6n^3 + 24n^2 + 36n + 24 - 15 - 20n - 10 - 15n^2 - 15n\right)\\
&= \frac{n(n+1)}{30}\left(6n^3 + 9n^2 + n - 1\right)
\end{aligned}$$

or

$$\begin{aligned}
(2n+1)(3n^2+3n-1) &= 6n^3 + 6n^2 - 2n + 3n^2 + 3n - 1\\
&= 6n^3 + 9n^2 + n - 1
\end{aligned}$$

$$\boxed{\forall n \in \mathbb{N}^*, \quad S_4(n) = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}}$$

EXERCICE 5

1) $(u_n) = (n!)$ est la suite définie sur $\mathbb{N}$: $u_0 = 1$ et $\forall n \in \mathbb{N},\ u_{n+1} = (n+1)u_n$.

Ou encore : $u_n = \prod_{k=1}^{n} k$

$0! = 1 \qquad 1! = 1 \qquad 2! = 2 \qquad 3! = 6 \qquad 4! = 24 \qquad 5! = 120 \qquad 6! = 720$

2)

```
4:     P = P*(k+1))                                # ou P = P*(n-k)
```

3)

```
def facto(n):
    P = 1
    for k in range(n):
        P = P*(k+1)                                # ou P = P*(n-k)
    return P
```

4)

```
def dernier(n):
    while n%10 == 0:
        n = n//10
    return n%10
```

5) *(ici j'ai changé l'énoncé original)*

```
c = 0
P = 1
for n in range(1, 1001):
    P = P*n                    # P va décrire les n! pour n allant de 1 à 1000
    if dernier(P) == 2:
        c += 1
print(c)
```