Question 1 :

Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = x\mathrm{e}^x$.
Une primitive F sur $\mathbb{R}$ de la fonction f est définie par :

A. $F(x) = \dfrac{x^2}{2}\mathrm{e}^x$

B $F(x) = (x-1)\mathrm{e}^x$

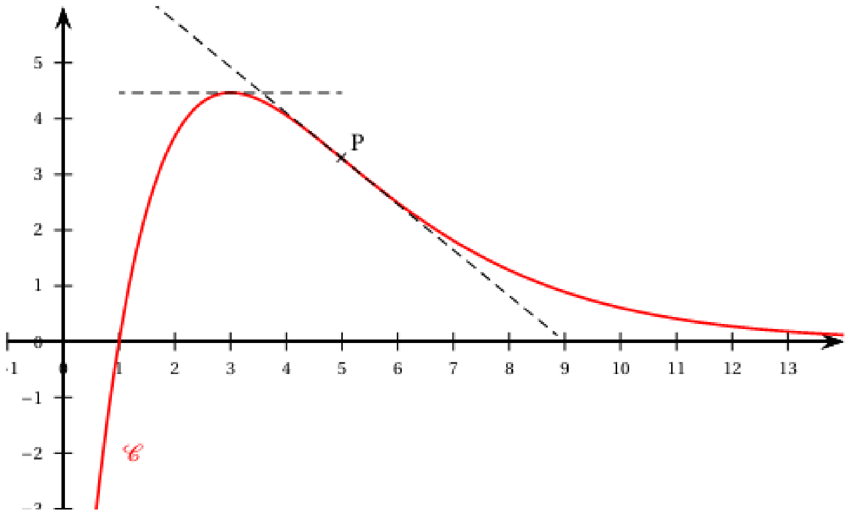
C. $F(x) = (x+1)\mathrm{e}^x$

D. $F(x) = \dfrac{2}{x}\mathrm{e}^{x^2}$.

Question 2 :

La courbe $\mathscr{C}$ ci-dessous représente une fonction f définie et deux fois dérivable sur $]0\,;\,+\infty[$. On sait que :

- le maximum de la fonction f est atteint au point d'abscisse 3 ;
- le point P d'abscisse 5 est l'unique point d'inflexion de la courbe $\mathscr{C}$.



On a :

A. pour tout $x \in]0\,;\,5[$, $f(x)$ et $f'(x)$ sont de même signe ;

B. pour tout $x \in]5\,;\,+\infty[$, $f(x)$ et $f'(x)$ sont de même signe ;

C. pour tout $x \in]0\,;\,5[$, $f'(x)$ et $f''(x)$ sont de même signe ;

D. pour tout $x \in]5\,;\,+\infty[$, $f(x)$ et $f''(x)$ sont de même signe.

Question 3 :

On considère la fonction g définie sur $[0\,;\,+\infty[$ par $g(t) = \dfrac{a}{b+\mathrm{e}^{-t}}$ où a et b sont deux nombres réels.
On sait que $g(0) = 2$ et $\lim\limits_{t\to+\infty} g(t) = 3$.
Les valeurs de a et b sont :

A. $a = 2$ et $b = 3$

B. $a = 4$ et $b = \dfrac{4}{3}$

C. $a = 4$ et $b = 1$

D. $a = 6$ et $b = 2$

Question 4 :

Alice dispose de deux urnes A et B contenant chacune quatre boules indiscernables au toucher.
L'urne A contient deux boules vertes et deux boules rouges.
L'urne B contient trois boules vertes et une boule rouge.
Alice choisit au hasard une urne puis une boule dans cette urne. Elle obtient une boule verte.
La probabilité qu'elle ait choisi l'urne B est :

A. $\dfrac{3}{8}$

B. $\dfrac{1}{2}$

C. $\dfrac{3}{5}$

D. $\dfrac{5}{8}$

Question 5 :

On pose $S = 1 + \dfrac{1}{2} + \dfrac{1}{3} + \dfrac{1}{4} + \ldots + \dfrac{1}{100}$.
Parmi les scripts Python ci-dessous, celui qui permet de calculer la somme S est :

a.
```
def somme_a() :
    S = 0
    for k in range(100) :
        S=1/(k+1)
    return S
```

b.
```
def somme_b() :
    S = 0
    for k in range(100) :
        S = S + 1/(k + 1)
    return S
```

c.
```
def somme_c() :
    k = 0
    while S < 100 :
        S = S+1/(k+1)
    return S
```

d.
```
def somme_d() :
    k = 0
    while k < 100 :
        S = S + 1/(k + 1)
    return S
```