T exp

Interrogation écrite du jeudi 12 janvier 2023 (30 minutes)

Numéro : ….. **Prénom et nom :** ……………………………………………… **Note : ….. / 20**

I. (5 points : 3 points + 2 points)

Dans cet exercice, on s'intéresse à deux jeux pour lesquels on dispose d'une urne contenant n boules indiscernables au toucher numérotées de 1 à n (n étant un entier naturel supérieur ou égal à 1).
Le joueur tire au hasard successivement, avec remise, deux boules dans l'urne. Il note les numéros des deux boules.
Dans le jeu 1, il gagne si l'un des deux numéros est pair ; il perd dans le cas contraire.
Dans le jeu 2, il gagne si les deux numéros sont congrus modulo 3 ; il perd dans le cas contraire.
Compléter les pointillés afin que les fonctions Python d'en-têtes `def jeu_1(n):` et `def jeu_2(n):` données dans les cadres ci-dessous permettent de simuler une partie de chacun des deux jeux.
On suppose que la fonction `randint` de la bibliothèque `random` a été préalablement importée.

```
def jeu_1(n):
    r1=randint(1,n)
    r2=randint(1,n)
    if ……………………….:
      print('gagné')
    else:
      print('perdu')
```

```
def jeu_2(n):
    r1=randint(1,n)
    r2=randint(1,n)
    if ……..==……..:
      print('gagné')
    else:
      print('perdu')
```

II. (5 points : 1 point + 1 point + 1 point + 1 point + 1 point)

Pour tout entier naturel $n \geqslant 1$, on note u_n l'entier naturel dont l'écriture en base dix comporte n fois le groupe 25.

On peut donc écrire $u_n = \overline{2525...25}^{(\text{dix})}$, le groupe de chiffres 25 étant répété n fois.
On considère la fonction Python d'en-tête `def rep_dc(n):` donnée dans le cadre ci-dessous qui prend pour argument un entier naturel n supérieur ou égal à 1 et qui a pour objectif de renvoyer la liste des entiers u_1, u_2, …, u_n dans cet ordre.
Compléter les pointillés. On pourra observer que pour tout entier naturel $n \geqslant 1$, on a $u_{n+1} = 100u_n + 25$.

```
def rep_dc(n):
    u=25
    L=[u]
    for _ in range(….,……):
        u=…………..:
        L.append(…)
    return …
```

III. (3 points : 1 point + 1 point + 1 point)

La fonction Python d'en-tête `def test_comp_modules(u,v):` donnée dans le cadre au verso prend pour arguments deux nombres complexes u et v.
Compléter les pointillés afin qu'elle renvoie `True` si $|u| \leqslant |v|$ et `False` sinon.

```
def test_comp_modules(u,v):
    if ............................ :
      return ........
    else:
      return ........
```

Que renvoie l'exécution de l'instruction

`test_comp_modules(1+3j,2-j) ?` ....

`test_comp_modules(5j,3+4i) ?` ....

IV. (7 points : 1°) 2 points ; 2°) 2 points ; 3°) 2 points ; 4°) 1 point)

Dans le plan complexe *P* muni d'un repère orthonormé direct $\left(\mathrm{O}, \vec{u}, \vec{v}\right)$, on donne les points A, B, C d'affixes respectives $z_{\mathrm{A}} = \mathrm{i} - 2$, $z_{\mathrm{B}} = 3 - 4\mathrm{i}$, $z_{\mathrm{C}} = 4 + 3\mathrm{i}$.
Faire un graphique soigné sur la feuille annexe.

1°) ....

....

....

....

2°) ....

....

....

....

....

3°) ....

....

....

....

....

Numéro : …..

Prénom et nom : ……………..……………………………….

Feuille annexe

IV. Questions

1°) Calculer les modules de z_B et z_C. Que peut-on dire des distances OB et OC ? Justifier.

2°) Calculer les affixes des vecteurs $\overrightarrow{AB}$ et $\overrightarrow{BC}$ puis calculer les distances AB et BC.
En déduire la nature du triangle ABC.

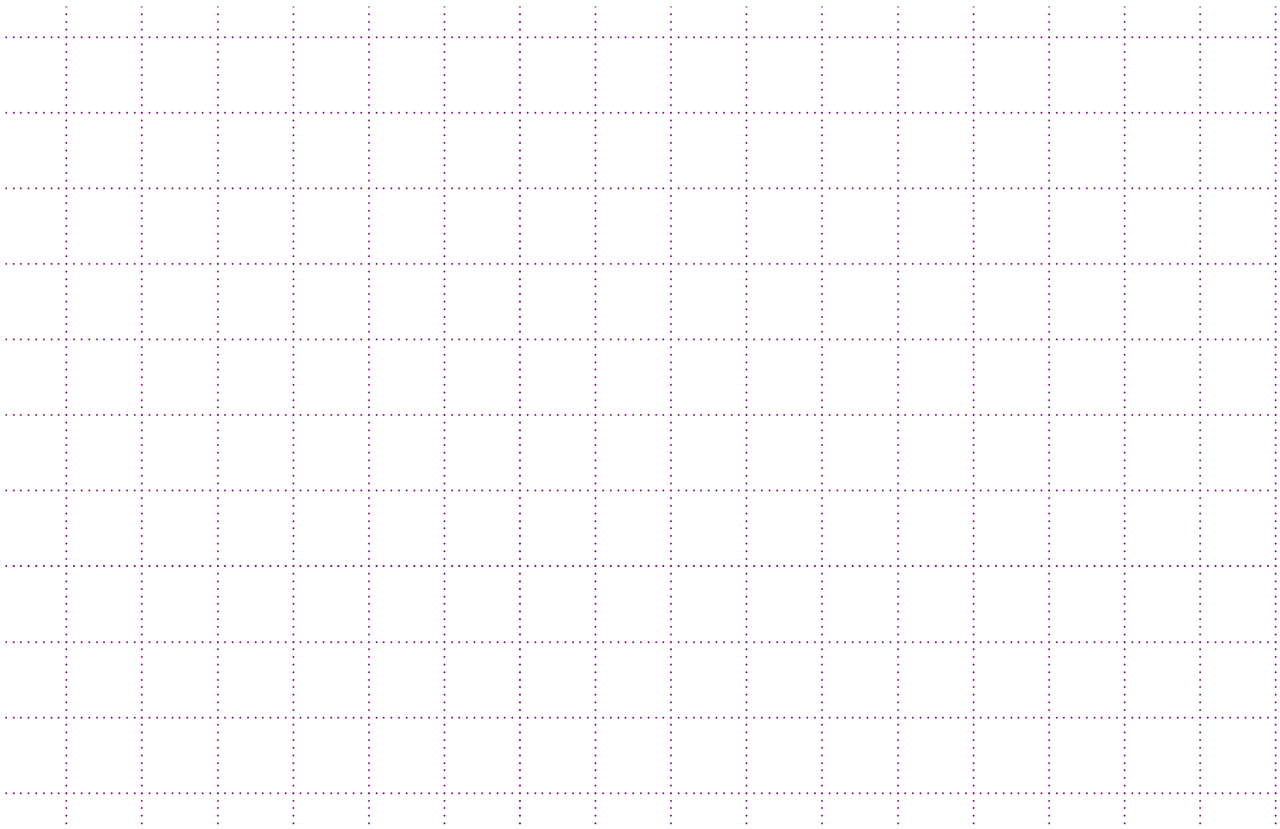
3°) Déterminer l'ensemble *E* des points M de *P*, d'affixe *z*, tels que $|z+2-i|=|z|$.
On attend une rédaction soignée selon le modèle suivant à recopier et compléter (ne rien écrire).

Soit M un point quelconque de *P*, d'affixe *z*. $M \in E \Leftrightarrow$ ……….. $\Leftrightarrow$ ……….. $\Leftrightarrow$ ……….. Conclusion : L'ensemble *E* est le / la …… .

Tracer *E* sur le graphique.

4°) Tracer sur le graphique l'ensemble *F* des points M de *P*, d'affixe *z*, tels que $|z|=|5|$.

Graphique de l'exercice IV



Corrigé de l'interrogation écrite du 12-1-2023

I.

Dans cet exercice, on s'intéresse à deux jeux pour lesquels on dispose d'une urne contenant n boules indiscernables au toucher numérotées de 1 à n (n étant un entier naturel supérieur ou égal à 1).
Le joueur tire au hasard successivement, avec remise, deux boules dans l'urne. Il note les numéros des deux boules.
Dans le jeu 1, il gagne si l'un des deux numéros est pair ; il perd dans le cas contraire.
Dans le jeu 2, il gagne si les deux numéros sont congrus modulo 3 ; il perd dans le cas contraire.
Compléter les pointillés afin que les fonctions Python d'en-têtes `def jeu_1(n):` et `def jeu_2(n):` données dans les cadres ci-dessous permettent de simuler une partie de chacun des deux jeux.
On suppose que la fonction `randint` de la bibliothèque `random` a été préalablement importée.

```
def jeu_1(n):
    r1=randint(1,n)
    r2=randint(1,n)
    if r1%2==0 or r2%2==0:
      print('gagné')
    else:
      print('perdu')
```

```
def jeu_2(n):
    r1=randint(1,n)
    r2=randint(1,n)
    if r1%3==r2%3:
      print('gagné')
    else:
      print('perdu')
```

Dans le cadre à droite, on peut écrire la condition sous la forme :

`r1%3==r2%3` (utilisation de la propriété : « deux entiers relatifs sont congrus modulo n, n étant un entier naturel supérieur ou égal à 2, si et seulement si ils ont le même reste dans la division euclidienne par n »)

ou (moins bien)

`(r1-r2)%3==0` (utilisation de la définition : « deux entiers relatifs sont congrus modulo n, n étant un entier naturel supérieur ou égal à 2, si et seulement si leur différence est divisible par n »).

II.

Pour tout entier naturel $n \geqslant 1$, on note u_n l'entier naturel dont l'écriture en base dix comporte n fois le groupe 25.

On peut donc écrire $u_n = \overline{2525...25}^{(\text{dix})}$, le groupe de chiffres 25 étant écrit n fois.
On considère la fonction Python d'en-tête `def rep_dc(n):` donnée dans le cadre ci-dessous qui prend pour argument un entier naturel n supérieur ou égal à 1 et qui a pour objectif de renvoyer la liste des entiers u_1, u_2, …, u_n dans cet ordre.
Compléter les pointillés. On pourra observer que pour tout entier naturel $n \geqslant 1$, on a $u_{n+1} = 100u_n + 25$.

```
def rep_dc(n):
    u=25
    L=[u]
    for _ in range(1,n):
        u=100*u+25:
        L.append(u)
    return L
```

Le tiret du bas _ remplace une variable de boucle.
Cela évite de donner un nom à la variable de boucle, qui n'intervient pas après.

Le nom de la fonction `rep_dc` vient de « répétition deux cinq ».

III.

La fonction Python d'en-tête `def test_comp_modules(u,v):` donnée dans le cadre au verso prend pour arguments deux nombres complexes *u* et *v*.
Compléter les pointillés afin qu'elle renvoie `True` si $|u| \leqslant |v|$ et `False` sinon.

```
def test_comp_modules(u,v):
    if abs(u)<=abs(v):
      return True
    else:
      return False
```

Que renvoie l'exécution de l'instruction

`test_comp_modules(1+3j,2-j) ?`   `False`

`test_comp_modules(5j,3+4i) ?`   `True`

IV.

Dans le plan complexe *P* muni d'un repère orthonormé direct $\left(\mathrm{O}, \vec{u}, \vec{v}\right)$, on donne les points A, B, C d'affixes respectives $z_{\mathrm{A}} = \mathrm{i} - 2$, $z_{\mathrm{B}} = 3 - 4\mathrm{i}$, $z_{\mathrm{C}} = 4 + 3\mathrm{i}$.
Faire un graphique soigné sur la feuille annexe.
1°) Calculer les modules de z_{B} et z_{C}. Que peut-on dire des distances OB et OC ? Justifier.
2°) Calculer les affixes des vecteurs $\overrightarrow{\mathrm{AB}}$ et $\overrightarrow{\mathrm{BC}}$ puis calculer les distances AB et BC.
En déduire la nature du triangle ABC.
3°) Déterminer l'ensemble *E* des points M de *P*, d'affixe *z*, tels que $|z + 2 - \mathrm{i}| = |z|$.
On attend une rédaction soignée en rédigeant selon le modèle suivant à recopier et compléter (ne rien écrire).

> Soit M un point quelconque de *P*, d'affixe *z*.
>
> $\mathrm{M} \in E \Leftrightarrow$ ………..
>
> $\Leftrightarrow$ ………..
>
> $\Leftrightarrow$ ………..
>
> Conclusion : L'ensemble *E* est le / la …… .

Tracer *E* sur le graphique.

4°) Tracer sur le graphique l'ensemble *F* des points M de *P*, d'affixe *z*, tels que $|z| = |5|$.

1°)

$|\ z_{\mathrm{B}}\ |=\sqrt{3^2+(-4)^2}=\sqrt{25}=5$

$|\ z_{\mathrm{C}}\ |=\sqrt{4^2+3^2}=\sqrt{25}=5$

On sait que :

$\mathrm{OB}=|\ z_{\mathrm{B}}\ |$ donc $\mathrm{OB}=5$;

$\mathrm{OC}=|\ z_{\mathrm{C}}\ |$ donc $\mathrm{OC}=5$.

On constate que $\mathrm{OB}=\mathrm{OC}$, c'est-à-dire que les distances OB et OC sont égales.

2°)

$z_{\overrightarrow{\mathrm{AB}}}=z_{\mathrm{B}}-z_{\mathrm{A}}=3-4\mathrm{i}-(\mathrm{i}-2)=5-5\mathrm{i}$

$z_{\overrightarrow{\mathrm{BC}}}=z_{\mathrm{C}}-z_{\mathrm{B}}=4+3\mathrm{i}-(3-4\mathrm{i})=1+7\mathrm{i}$

$\mathrm{AB}=|\ z_{\overrightarrow{\mathrm{AB}}}\ |=\sqrt{5^2+(-5)^2}=\sqrt{50}=5\sqrt{2}$

$\mathrm{BC}=|\ z_{\overrightarrow{\mathrm{BC}}}\ |=\sqrt{1^2+7^2}=\sqrt{50}=5\sqrt{2}$

On pourrait s'arrêter à $\sqrt{50}$ dans les deux cas.

On constate que $\mathrm{AB}=\mathrm{BC}$. On en déduit que le triangle ABC est isocèle en B.

3°)

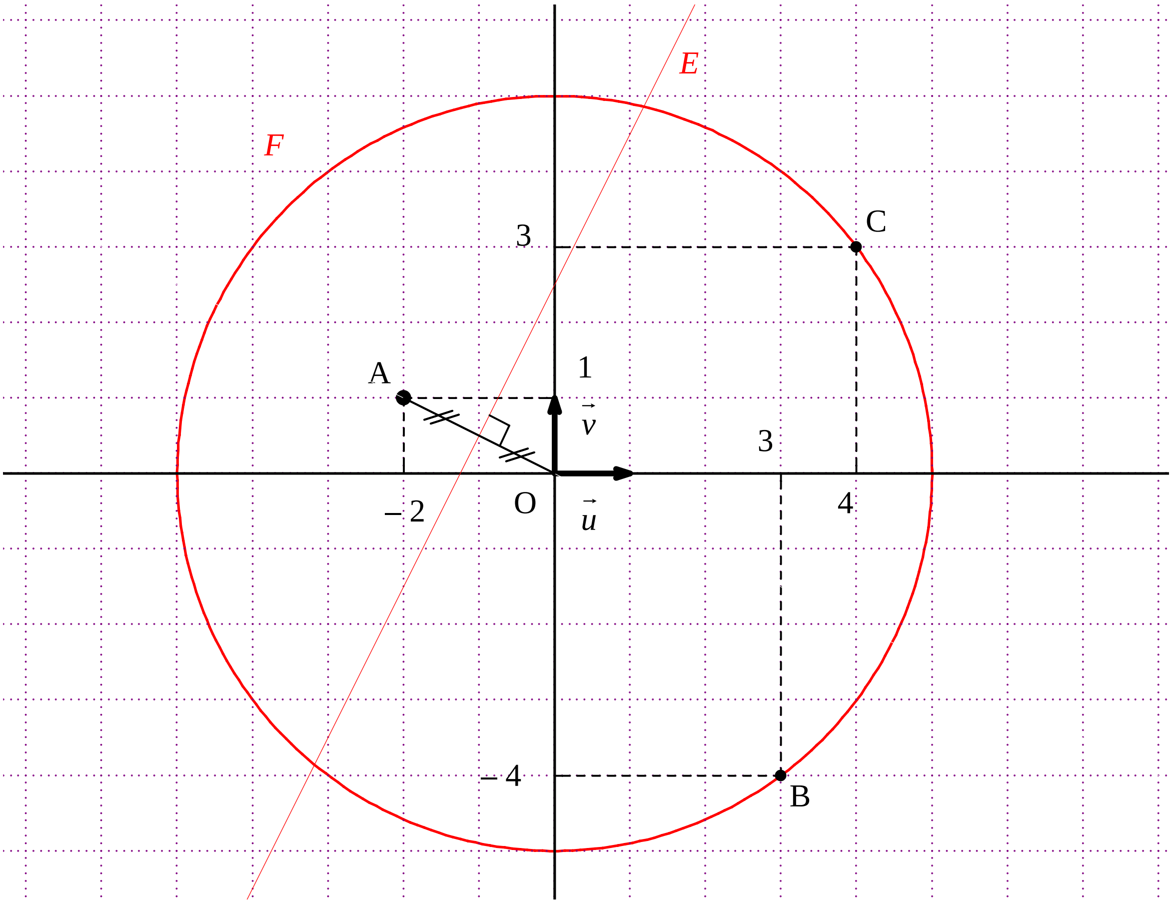
Soit M un point quelconque de *P*, d'affixe *z*.

$$\begin{aligned}
\mathrm{M}\in E &\Leftrightarrow |\ z-(\mathrm{i}-2)\ |=|\ z\ |\\
&\Leftrightarrow |\ z-z_{\mathrm{A}}\ |=|\ z\ |\\
&\Leftrightarrow \mathrm{AM}=\mathrm{OM}\\
&\Leftrightarrow \mathrm{MA}=\mathrm{MO}\quad \text{(ligne facultative)}
\end{aligned}$$

Conclusion : L'ensemble *E* est la médiatrice du segment $[\mathrm{OA}]$.

Sur le graphique, on marque le codage pour la médiatrice.

4°) L'ensemble *F* est le cercle de centre O et de rayon 5.
F passe par les points B et C car on a démontré à la question 1°) que $\mathrm{OB}=\mathrm{OC}=5$.



E
F
C
3
A
1
$\vec{v}$
3
O
− 2
$\vec{u}$
4
− 4
B