

RÉPUBLIQUE TUNISIENNE ★★★ MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2022 ÉPREUVE PRATIQUE D'INFORMATIQUE Sections : Mathématiques, Sciences expérimentales et Sciences techniques	NOUVEAU RÉGIME
	Coefficient de l'épreuve : 0.5	Durée : 1h

Important :

1. Une solution modulaire au problème posé est exigée.
2. Dans le répertoire **Bac2022**, créez un dossier de travail ayant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Le nombre ondulant

Un nombre **N** est dit **ondulant** s'il est formé de trois chiffres au minimum et qui est de la forme **ababab...** avec **a ≠ b**.

Exemples :

- 101, 2525, 56565 et 1717171 sont des nombres ondulants.
- 12345 et 808008 ne sont pas des nombres ondulants

Pour vérifier si un entier naturel **N** ($N \geq 100$) est un nombre **ondulant** ou non, on se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : "**Nombre ondulant**"
- Un label demandant la saisie d'un nombre "**Introduire un entier ≥ 100 :**"
- Une zone de saisie permettant la saisie du nombre
- Un bouton intitulé "**Vérifier**"
- Un label pour afficher le message adéquat

Nombre ondulant

Introduire un entier ≥ 100 :

Vérifier

Travail demandé :

- 1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterfaceOndulant**".
- 2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**NbrOndulant**".
- 3) Développer, dans le programme "**NbrOndulant**", une fonction **Ondulant(N)** qui permet de vérifier si un entier **N** est ondulant ou non.
- 4) Dans le programme "**NbrOndulant**" :
 - ajouter les instructions permettant d'appeler l'interface graphique intitulée "**InterfaceOndulant**" en exploitant l'annexe ci-après.
 - développer un module "**Play**", qui s'exécute suite à un clic sur le bouton "**Vérifier**", permettant de récupérer l'entier **N** saisi, puis d'exploiter la fonction "**Ondulant**" afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface "**InterfaceOndulant**".

N.B. : l'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution :

<p>Nombre ondulant</p> <p>Introduire un entier >= 100 : <input type="text" value="27"/></p> <p><input type="button" value="Vérifier"/></p> <p>Veuillez introduire un nombre >= 100</p>	<p>Nombre ondulant</p> <p>Introduire un entier >= 100 : <input type="text" value="56565"/></p> <p><input type="button" value="Vérifier"/></p> <p>56565 est ondulont</p>
---	---

<p>Nombre ondulant</p> <p>Introduire un entier >= 100 : <input type="text" value="808008"/></p> <p><input type="button" value="Vérifier"/></p> <p>808008 n'est pas ondulont</p>

<p>Annexe</p> <pre>from PyQt5.uic import loadUi from PyQt5.QtWidgets import QApplication app = QApplication([]) windows = loadUi ("Nom_Interface.ui") windows.show() windows.Nom_Bouton.clicked.connect (Nom_Module) app.exec_()</pre>

Grille d'évaluation

Tâches	Nombre de points
Conception de l'interface " InterfaceOndulant "	4 pts
Création et enregistrement du programme " NbrOndulant "	1 pt
Développement de la fonction " Ondulant "	6 pts
Ajout des instructions : <ul style="list-style-type: none">▪ de l'interface "InterfaceOndulant"▪ du module "Play"	2 pts 4 pts
Importation des bibliothèques nécessaires, modularité et cohérence	3 pts

<p>RÉPUBLIQUE TUNISIENNE ★★★ MINISTÈRE DE L'ÉDUCATION</p>	EXAMEN DU BACCALAURÉAT SESSION 2022	NOUVEAU RÉGIME
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et Sciences techniques	
	Coefficient de l'épreuve : 0.5	Durée : 1h

Important :

1. Une solution modulaire au problème posé est exigée.
2. Dans le répertoire **Bac2022**, créez un dossier de travail ayant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Le nombre super-pairplus

Un nombre **N** est dit **super-pairplus** s'il vérifie les trois conditions suivantes :

- **Condition 1 :** N est pair
- **Condition 2 :** N est formé uniquement par des chiffres pairs
- **Condition 3 :** tous les diviseurs de N autre que **1** sont des entiers pairs.

Exemples :

- $N = 64$ est un entier **super-pairplus** car 64 vérifie les trois conditions :
 - **Condition 1 :** 64 est pair
 - **Condition 2 :** 64 est formé uniquement par des chiffres pairs (6 et 4)
 - **Condition 3 :** les diviseurs de 64 autre que **1** (2, 4, 8, 16, 32 et 64) sont des entiers pairs.
- $N = 28$ n'est pas un entier **super-pairplus** car une des trois conditions (**condition 3**) n'est pas vérifiée :
 - **Condition 1 :** 28 est pair
 - **Condition 2 :** 28 est formé uniquement par des chiffres pairs (2 et 8)
 - **Condition 3 :** les diviseurs de 28 autre que **1** (2, 4, 7, 14 et 28) ne sont pas tous des entiers pairs. En effet, 7 est impair

Pour vérifier si un entier naturel **N** ($N > 0$) est un nombre **super-pairplus** ou non, on se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : "**Nombre super-pairplus**"
- Un label demandant la saisie d'un nombre "**Introduire un entier > 0 :**"
- Une zone de saisie permettant la saisie du nombre
- Un bouton intitulé "**Vérifier**"
- Un label pour afficher le message adéquat

Nombre super-pairplus

Introduire un entier > 0 :

Vérifier

Travail demandé :

- 1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterfaceSuperPairplus**".
- 2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**NbrSuperPairplus**".
- 3) Développer, dans le programme "**NbrSuperPairplus**", une fonction **SuperPairplus (N)** qui permet de vérifier si un entier **N** est super-pairplus ou non.

4) Dans le programme " NbrSuperPairplus" :

- ajouter les instructions permettant d'appeler l'interface graphique intitulée "**InterfaceSuperPairplus**" en exploitant l'annexe ci-après.
- développer un module "**Play**", qui s'exécute suite à un clic sur le bouton "**Vérifier**", permettant de récupérer l'entier **N** saisi, puis d'exploiter la fonction "**SuperPairplus**" afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface "**InterfaceSuperPairplus**".

N.B. : l'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution :

Nombre super-pairplus

Introduire un entier > 0 :

Veuillez introduire un nombre > 0

Nombre super-pairplus

Introduire un entier > 0 :

64 est super-pairplus

Nombre super-pairplus

Introduire un entier > 0 :

28 n'est pas super-pairplus

Annexe

```
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QApplication
.....
.....
app = QApplication([])
windows = loadUi ("Nom_Interface.ui")
windows.show()
windows.Nom_Bouton.clicked.connect (Nom_Module)
app.exec_()
```

Grille d'évaluation

Tâches	Nombre de points
Conception de l'interface " InterfaceSuperPairplus "	4 pts
Création et enregistrement du programme " NbrSuperPairplus "	1 pt
Développement de la fonction " SuperPairplus "	6 pts
Ajout des instructions : <ul style="list-style-type: none"> ▪ de l'interface "InterfaceSuperPairplus" ▪ du module "Play" 	2 pts 4 pts
Importation des bibliothèques nécessaires, modularité et cohérence	3 pts

<p>RÉPUBLIQUE TUNISIENNE ★★★ MINISTÈRE DE L'ÉDUCATION</p>	EXAMEN DU BACCALAURÉAT SESSION 2022	NOUVEAU RÉGIME
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et Sciences techniques	
	Coefficient de l'épreuve : 0.5	Durée : 1h

Important :

1. Une solution modulaire au problème posé est exigée.
2. Dans le répertoire **Bac2022**, créez un dossier de travail ayant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Le nombre semi-premier

Un nombre **N** est dit **semi-premier** lorsqu'il est égal au produit de **deux nombres premiers** non nécessairement distincts. C'est-à-dire $N = k*k$ avec **k** est un nombre premier ou $N = k*j$ avec **k** et **j** sont deux nombres **premiers**.

Exemples :

- **6 est un nombre semi-premier** car $6 = 2 \times 3$ avec 2 et 3 sont deux nombres premiers.
- **25 est un nombre semi-premier** car $25 = 5 \times 5$ avec 5 est un nombre premier.
- **831 est un nombre semi-premier** car $831 = 3 \times 277$ avec 3 et 277 sont deux nombres premiers
- **8 n'est pas un nombre semi-premier**, car $8 = 2 \times 4$ avec 4 n'est pas un nombre premier.

Pour vérifier si un entier naturel **N** ($N > 2$) est un nombre **semi-premier** ou non, on se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : "**Nombre semi-premier**"
- Un label demandant la saisie d'un nombre : "**Introduire un entier > 2 :**"
- Une zone de saisie permettant la saisie du nombre
- Un bouton intitulé "**Vérifier**"
- Un label pour afficher le message adéquat

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

Travail demandé :

- 1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterfaceSemiPremier**".
- 2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**NbrSemiPremier**".
- 3) Développer, dans le programme "**NbrSemiPremier**", une fonction **SemiPremier(N)** qui permet de vérifier si un entier **N** est semi-premier ou non.

4) Dans le programme "NbrSemiPremier" :

- ajouter les instructions permettant d'appeler l'interface graphique intitulée "**InterfaceSemiPremier**" en exploitant l'annexe ci-après.
- développer un module "**Play**", qui s'exécute suite à un clic sur le bouton "**Vérifier**", permettant de récupérer l'entier **N** saisi, puis d'exploiter la fonction "**SemiPremier**" afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface "**InterfaceSemiPremier**".

N.B. :

- Le candidat est appelé à développer un module qui permet de vérifier la primalité d'un entier **sans faire recours** à des fonctions prédéfinies telles que **isprime()**.
- l'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution :

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

Veuillez introduire un nombre > 2

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

831 est semi-premier

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

8 n'est pas semi-premier

Annexe

```
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QApplication
.....
.....
app = QApplication([])
windows = loadUi ("Nom_Interface.ui")
windows.show()
windows.Nom_Bouton.clicked.connect (Nom_Module)
app.exec_()
```

Grille d'évaluation

Tâches	Nombre de points
Conception de l'interface " InterfaceSemiPremier "	4 pts
Création et enregistrement du programme "NbrSemiPremier"	1 pt
Développement de la fonction " SemiPremier "	6 pts
Ajout des instructions :	
■ de l'interface " InterfaceSemiPremier "	2 pts
■ du module " Play "	4 pts
Importation des bibliothèques nécessaires, modularité et cohérence	3 pts

<p>RÉPUBLIQUE TUNISIENNE ★★★ MINISTÈRE DE L'ÉDUCATION</p>	EXAMEN DU BACCALAURÉAT SESSION 2022	NOUVEAU RÉGIME
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et Sciences techniques	
	Coefficient de l'épreuve : 0.5	Durée : 1h

Important :

1. Une solution modulaire au problème posé est exigée.
2. Dans le répertoire **Bac2022**, créez un dossier de travail ayant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Le nombre semi-premier

Un nombre **N** est dit **semi-premier** lorsqu'il est égal au produit de **deux nombres premiers** non nécessairement distincts. C'est-à-dire $N = k*k$ avec **k** est un nombre premier ou $N = k*j$ avec **k** et **j** sont deux nombres **premiers**.

Exemples :

- **6 est un nombre semi-premier** car $6 = 2 \times 3$ avec 2 et 3 sont deux nombres premiers.
- **25 est un nombre semi-premier** car $25 = 5 \times 5$ avec 5 est un nombre premier.
- **831 est un nombre semi-premier** car $831 = 3 \times 277$ avec 3 et 277 sont deux nombres premiers
- **8 n'est pas un nombre semi-premier**, car $8 = 2 \times 4$ avec 4 n'est pas un nombre premier.

Pour vérifier si un entier naturel **N** ($N > 2$) est un nombre **semi-premier** ou non, on se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : "**Nombre semi-premier**"
- Un label demandant la saisie d'un nombre : "**Introduire un entier > 2 :**"
- Une zone de saisie permettant la saisie du nombre
- Un bouton intitulé "**Vérifier**"
- Un label pour afficher le message adéquat

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

Travail demandé :

- 1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterfaceSemiPremier**".
- 2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**NbrSemiPremier**".
- 3) Développer, dans le programme "**NbrSemiPremier**", une fonction **SemiPremier(N)** qui permet de vérifier si un entier **N** est semi-premier ou non.

4) Dans le programme "NbrSemiPremier" :

- ajouter les instructions permettant d'appeler l'interface graphique intitulée "**InterfaceSemiPremier**" en exploitant l'annexe ci-après.
- développer un module "**Play**", qui s'exécute suite à un clic sur le bouton "**Vérifier**", permettant de récupérer l'entier **N** saisi, puis d'exploiter la fonction "**SemiPremier**" afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface "**InterfaceSemiPremier**".

N.B. :

- Le candidat est appelé à développer un module qui permet de vérifier la primalité d'un entier **sans faire recours** à des fonctions prédéfinies telles que **isprime()**.
- l'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution :

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

Veuillez introduire un nombre > 2

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

831 est semi-premier

Nombre semi-premier

Introduire un entier > 2 :

Vérifier

8 n'est pas semi-premier

Annexe

```
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QApplication
.....
.....
app = QApplication([])
windows = loadUi ("Nom_Interface.ui")
windows.show()
windows.Nom_Bouton.clicked.connect (Nom_Module)
app.exec_()
```

Grille d'évaluation

Tâches	Nombre de points
Conception de l'interface " InterfaceSemiPremier "	4 pts
Création et enregistrement du programme "NbrSemiPremier"	1 pt
Développement de la fonction " SemiPremier "	6 pts
Ajout des instructions :	
■ de l'interface " InterfaceSemiPremier "	2 pts
■ du module " Play "	4 pts
Importation des bibliothèques nécessaires, modularité et cohérence	3 pts