

RÉPUBLIQUE TUNISIENNE ----- MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2025	
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et Sciences techniques	
	Coefficient de l'épreuve : 0.5	Durée : 1h

Important :

Le candidat est appelé à créer, dans le répertoire **BAC2025**, un dossier de travail ayant comme nom son numéro d'inscription (6 chiffres) et dans lequel il doit enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Nombres ronds

On se propose de concevoir une interface graphique permettant de saisir deux entiers **A** et **B** tels que $(10 \leq A \leq 30)$ et $(A < B \leq 99)$, puis d'afficher tous les nombres ronds compris entre **A** et **B**.

Un nombre est dit **rond** si sa conversion à la base 2 (base binaire) contient autant de 1 que de 0 (le nombre des 0 égal à celui des 1).

Exemples :

- Pour **A = 20 et B = 40** :

Le programme affiche : **Les nombres ronds sont : 35-37-38**

En effet, - la conversion binaire de $35_{(10)}$ vaut $100011_{(2)}$ qui contient autant de 1 que de 0.

- la conversion binaire de $37_{(10)}$ vaut $100101_{(2)}$ qui contient autant de 1 que de 0.

- la conversion binaire de $38_{(10)}$ vaut $100110_{(2)}$ qui contient autant de 1 que de 0.

- Pour **A = 15 et B = 30** :

Il n'existe aucun nombre rond compris entre **15** et **30**, d'où :

Le programme affiche : **Aucun nombre rond**

Travail demandé

- 1) Créer l'interface graphique illustrée dans la figure **Fig-1** et l'enregistrer sous le nom **Interface**.

Cette interface contient les éléments suivants :

- Un label contenant le texte : "**Nombres ronds**".
- Un label contenant le texte : "**A =**".
- Une zone de saisie pour la saisie d'un entier **A**.
- Un label contenant le texte : "**B =**".
- Une zone de saisie pour la saisie d'un entier **B**.
- Un bouton intitulé "**Afficher**".
- Un label dédié à l'affichage.

Fig-1

- 2) Créer un programme python et l'enregistrer sous le nom **NombreRond**, dans lequel on demande :
- a) d'implémenter l'algorithme suivant de la fonction **Conv_binaire** qui permet de convertir un entier **N** en binaire.

Fonction Conv_binaire (N : Entier) : Chaîne de caractères

DEBUT

Si (N=0) **Alors**

ch ← "0"

Sinon

ch ← ""

Tant que N ≠ 0 **Faire**

r ← N Mod 2

N ← N Div 2

ch ← Convch (r) + ch

Fin Tant que

Retourner ch

FIN

T.D.O.L

Objet	Type/Nature
ch	Chaîne de caractères
r	Entier

- b) de développer une fonction **Verif(N)** qui permet de vérifier si un entier **N** est rond, en utilisant la fonction **Conv_binaire** précédente.
- c) de développer une fonction **Ronds (A, B)** pour former une chaîne de caractères contenant les nombres ronds, de l'intervalle [A..B], séparés par le caractère "-", en utilisant la fonction **Verif**.
- d) de développer un module **Play**, qui s'exécute suite au clic sur le bouton "**Afficher**", permettant :
- de récupérer les valeurs saisies des entiers **A** et **B**, de s'assurer de leurs validité et d'afficher, le cas échéant, le message adéquat via le label dédié à l'affichage, comme illustré dans les figures **Fig-2** et **Fig-3**.
 - d'exploiter la fonction **Ronds** afin d'afficher le message adéquat via le label dédié à l'affichage, comme illustré dans les figures **Fig-4** et **Fig-5**.
- e) d'exploiter l'annexe présentée ci-après tout en apportant les modifications nécessaires à l'intégration de l'interface graphique **Interface**.

Nombres ronds

A =

B =

A doit être dans [10..30] !

Fig-2

Nombres ronds

A =

B =

B doit être dans [20..99] !

Fig-3

Nombres ronds

A =

B =

Les nombres ronds sont: 35-37-38

Fig-4

Nombres ronds

A =

B =

Aucun nombre rond

Fig-5

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
1) Création de l'interface Interface .	3
2) Création du programme Nombre_Rond .	17 =
a) Implémentation de la fonction Conv_binaire	4
b) Développement de la fonction Verif	3
c) Développement de la fonction Ronds	2.5
d) Développement du module Play .	6
e) Exploitation de l'annexe.	1.5