

RÉPUBLIQUE TUNISIENNE *** Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT	SESSION 2024
	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 **Bac2024**, 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 Primaires

On se propose de concevoir une interface graphique permettant de saisir deux entiers **N** ($10 \leq N \leq 50$) et **M** ($N < M \leq 200$), puis d'afficher tous les nombres **primaires** compris entre **N** et **M**.

Un nombre **P** est dit **primaire** s'il existe un nombre premier **X** tel que $P = X^k$ avec **k** un entier strictement positif.

Un nombre **premier** est un entier strictement positif qui admet exactement deux diviseurs positifs distincts 1 et lui-même.

Exemples :

- **144** n'est pas un nombre **primaire**. En effet, $144 = 12^2$ et **12** n'est pas un nombre premier.
- **169** est un nombre **primaire**. En effet, $169 = 13^2$ et **13** est un nombre premier.

Travail demandé

- Créer l'interface graphique illustrée dans la figure **Fig1** et l'enregistrer sous le nom **InterfacePrimaire**. Cette interface contient les éléments suivants :

- Un label contenant le texte "**Nombres Primaires**",
- Un label contenant le texte "**N**=",
- Une zone de saisie pour la saisie d'un entier **N**,
- Un label contenant le texte "**M**",
- Une zone de saisie pour la saisie d'un entier **M**,
- Un label pour afficher un message,
- Un bouton intitulé "**Afficher**".

Fig1

- Créer un programme en Python et l'enregistrer sous le nom **Primaire** dans lequel on demande :
 - d'implémenter l'algorithme suivant de la fonction **Premier(X)** qui vérifie si un entier strictement positif **X** est premier.

Fonction Premier (*X : Entier*) : *Booléen*

DEBUT

TEST \leftarrow Vrai

Si $X = 1$ Alors TEST \leftarrow Faux

Sinon Si $(X > 3)$ Alors

K \leftarrow 2

Tant que TEST Et $(K \leq X \text{ Div } 2)$ Faire

Si $X \text{ Mod } K = 0$ alors TEST \leftarrow Faux

Sinon $K = K + 1$

Fin Tant que

Retourner TEST

FIN

T.D.O.L

Objet	Type/Nature
K	Entier
TEST	Booléen

- b. de développer une fonction booléenne nommée **Vérifier(P)**, qui exploite la fonction **Premier**, afin de vérifier si l'entier **P** est un nombre **primaire**.
- c. de développer une fonction nommée **Chercher(N, M)**, qui exploite la fonction **Vérifier**, pour former une chaîne de caractères contenant tous les nombres **primaires** compris entre **N** et **M** séparés par le caractère "-" (voir **Fig4**).
- d. de développer un module **Play** qui s'exécute suite à un clic sur le bouton "**Afficher**" permettant :
 - de récupérer les valeurs des entiers **N** et **M** saisis et de s'assurer de leurs validités afin d'afficher le message adéquat via le **label** dédié à l'affichage, comme illustré dans la figure **Fig2**.
 - d'exploiter la fonction **Chercher** pour afficher le message adéquat via le **label** dédié à l'affichage, comme illustré dans les figures **Fig3** et **Fig4**.
- e. d'exploiter l'annexe présentée ci-après tout en apportant les modifications nécessaires à l'intégration de l'interface graphique **InterfacePrimaire**.

Nombres Primaires

N= **M=**

Afficher

Veuillez respecter : $10 \leq N \leq 50$ et $N < M \leq 200$

Fig2

Nombres Primaires

N= **M=**

Afficher

Aucun nombre primaire entre 35 et 45

Fig3

Nombres Primaires

N= **M=**

Afficher

Nombre(s) primaire(s) : 16-25-27-32-49-64-81-121-125-128-169

Fig4

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 InterfacePrimaire .	3
2. Création du programme Primaire .	17 =
a. Implémentation de la fonction Premier .	3
b. Développement de la fonction Vérifier .	4.5
c. Développement de la fonction Chercher .	4
d. Développement du module Play .	4
e. Exploitation de l'annexe.	1.5