

Examen SEMESTRE 1**Matière : INFORMATIQUE****Classes : 1^{ère} Année BG****Durée : 2 heures****EXERCICE 1:**

Soit le code Python suivant:

```

b=int(input("donner la valeur de b"))
m=int(input("donner la valeur de m"))
r = 1
while m > 0:
    if m % 2 == 1:
        r = r * b
    b = b * b
    m = m // 2
print(r)

```

1/ Quel est le résultat affiché:

A. Si b=2 et m=3

B. Si b=3 et m=3

2/ Quel est le rôle de ce programme.

EXERCICE 2:On considère deux suites numériques U_n et V_n telles que pour n strictement supérieur à 2 :

$$\begin{cases} U_n = U_{n-1} + U_{n-2} \\ \text{et} \\ V_n = \frac{U_n}{U_{n-1}} \end{cases}$$

Avec $U_0 = U_1 = 1$ La suite V_n tend vers une limite appelée nombre d'or.Ecrire un programme Python permettant de calculer et d'afficher une approximation du nombre d'or avec une précision $\varepsilon < 1E-6$ à saisir.On arrête les calculs quand : $\left| \frac{V_{n+1} - V_n}{V_n} \right| < \varepsilon$

EXERCICE 3:

Un nombre entier est dit 'Stepping Number' si la différence en valeur absolue entre tous les chiffres adjacents de ce même nombre est toujours égale à 1. Un tel nombre ne peut pas être composé par un seul chiffre, il doit donc être composé par au moins deux chiffres. Par exemple, 45 et 3434567 sont des 'Stepping Number'. Par contre, le nombre 8769 n'est pas un 'Stepping Number' $|8-7|=1$, $|7-6|=1$ alors que $|6-9|=3$.

Votre travail consiste à écrire un programme PYTHON qui permet de:

1. Saisir un entier N tel que $10 \leq N \leq 10^5$
2. Calculer et afficher tous les 'Stepping Number' de l'intervalle $[10, N]$.

à ne pas utiliser les transformations d'entiers en chaîne de caractères et vice versa.

EXERCICE 4:

Un distributeur de banque dispose de billets de 10, 20, 50, 100, 200 et 500 euros. Le client qui se présente demande de retirer une somme d'argent. Le distributeur la lui fournit avec une répartition minimale de billets (c'est à dire propose le moins de billets possible). Si le montant n'est pas multiple des billets disponibles le distributeur doit répondre par le message :

"Coupures non disponibles pour vous verser cette somme".

On suppose que le distributeur dispose d'un stock infini de chaque type de billets.

Ecrire un code Python qui demande la somme voulue et propose un nombre de billets totalisant la somme demandée.

On demande de respecter les étapes suivantes :

1. Créer une liste billets contenant les différents types de billets par ordre croissant.
2. Créer une liste nbre de longueur 6, initialisé par des zéros servant à enregistrer le nombre de billets à fournir de chaque type.
3. Saisir la somme S à retirer.
4. Remplir la liste nbre contenant la répartition selon le type des billets de la liste billets.
Exemple : La liste obtenue $[2, 0, 3, 0, 0, 1]$ représente deux billets de 10, trois billets de 50 et un billet de 500

5. Afficher :

a. "Coupures non disponibles pour vous verser cette somme, veuillez taper un autre montant" dans le cas où le montant est impossible.

b. la répartition en billets selon la liste des billets disponibles.

Exemple d'exécution :

```
>>> donner le montant à retirer : 2552
```

Coupures indisponible pour vous verser cette somme. Veuillez taper un autre montant : 2580

1 billet(s) de 10, 1 billet(s) de 20, 1 billet(s) de 50, 5 billet(s) de 500.