Lycée Béja Nord	**Sujet Pratique n°1**	Durée : 1 Heure
Classes : 4ème SC&M	Matière : Informatique — Enseignant : Kais TABOUBI	Date : **-**-2023

Important

1. *Une solution modulaire au problème posé est exigée.*

2. *Dans le répertoire* ***Bac2023****, créez un dossier de travail ayant comme nom votre numéro d'inscription (****6chiffres****) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.*

Dans le 100ème anniversaire, une entreprise commerciale décide, à la caisse, d'offrir aux clients leurs achats. Pour cela, et suite aux règlements de l'offre, le caissier demande du client son numéro de téléphone **N** (contient **8** chiffres), dont son premier doit être un parmi la liste suivante (**2, 3, 4, 5** et **9**).

On se propose d'afficher si le client qui a le numéro de téléphone **N** est gagnant ou non. Un client est déclaré gagnant si le **carré** du chiffre de chance **CC** de son numéro de téléphone existe dans un tableau **T** déjà rempli aléatoirement par **8** entiers appartenant à l'intervalle [**1, 99**].

Un chiffre de chance **CC** relatif à un numéro de téléphone est calculé en additionnant de façon répétitive tous les chiffres qui composent le numéro de téléphone jusqu'à obtenir un seul chiffre.

Exemples :

❶ Pour le numéro de téléphone **N** et le tableau **T** suivants: **N = 55 405 103** **T**

13	7	19	25	93	64	7	1
0	1	2	3	4	5	6	7

Le programme affiche " *Le client ayant le numéro de téléphone* **55 405 103** ***a gagné un chariot gratuit*** ".

En effet, le **CC** du client a été obtenu en additionnant les chiffres de son numéro de téléphone jusqu'à obtenir un seul chiffre c'est à dire **:** * **5+5+4+0+5+1+0+3 = 23**

** **2+3 =5.**

Le carré du chiffre **5, (5^2=25),** figure dans le tableau **T**, donc c'est un client gagnant.

❷ Pour le numéro de téléphone **N** et le tableau **T** suivants: **N = 99 678 987** **T**

8	54	96	38	54	5	11	54
0	1	2	3	4	5	6	7

Le programme affiche " *Le client ayant le numéro de téléphone* **99 678 987** ***doit payer les achats du chariot*** ".

En effet, le **CC** du client a été obtenu en additionnant les chiffres de son numéro de téléphone jusqu'à obtenir un seul chiffre c'est à dire **:** * **9+9+6+7+8+9+8+5 = 55**

** **5+5 =10**

*** **1+0 =1.**

Le carré du chiffre **1, (1^2=1),** ne figure pas dans le tableau **T**, donc c'est un client perdant.

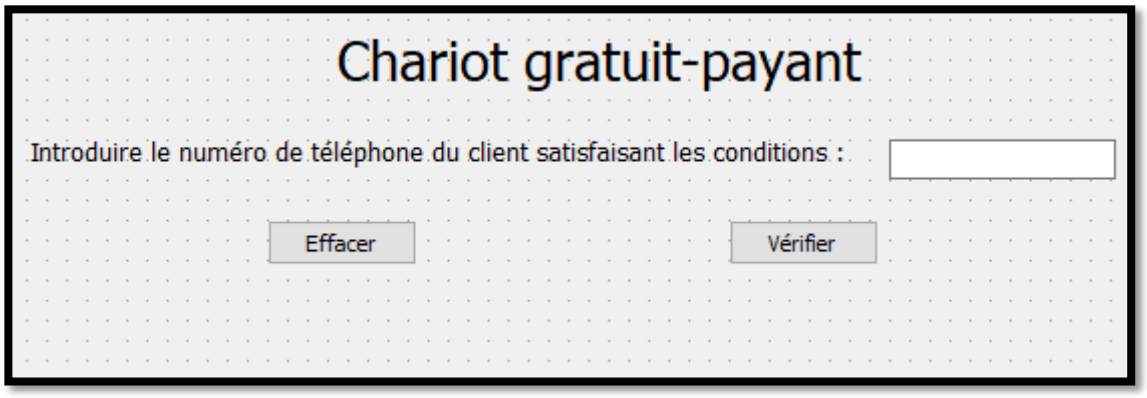
Ci-après, la procédure **REMP_AFFICHE** à exploiter pour résoudre le problème posé, qui permet de remplir aléatoirement le tableau **T**, de type TAB, par **8** entiers dans l'intervalle [**1 , 99**] afin de les afficher sur la console.

```
Procédure REMP_AFFICHE (@ T : TAB)
  Début
    Pour i de 0 à 7 faire
      T[i] ← Aléa(1 , 99)
      Ecrire (T[i] , " | ")
    Fin Pour
  Fin
```

NB : l'affichage des éléments du tableau **T** dans la console vous aide pour tester votre programme lors de la saisie du numéro de téléphone du client (gain ou perd).

On se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : **"Chariot gratuit-payant"**
- Un label demandant la saisie du nombre :
  "Introduire le numéro de téléphone du client satisfaisant les conditions :"
- Une zone de saisie permettant la saisie du nombre.
- Un bouton intitulé **" Valider"**
- Un bouton intitulé **"Effacer"** *(permettant d'effacer la zone de saisie et le label d'affichage)*
- Un label pour afficher le message adéquat.



Travail demandé :

1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom **"InterCHARIOT"**.

2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom **"PgmeCHAR".**

3) Implémenter la procédure **REMP_AFFICHE** et l'appeler dans le programme principal.

4) Développer dans le programme **" PgmeCHAR"**, une fonction **GainPerd (N)** qui permet de vérifier si le numéro de téléphone du client **N** est gagnant ou non.

5) Dans le programme **" PgmeCHAR"** :

   - Ajouter les instructions permettant d'appeler l'interface graphique intitulée **" InterCHARIOT"** en exploitant **l'annexe** ci-après.
   - Développer un module **"Play",** qui s'exécute suite à un clic sur le bouton **"Valider",** permettant de récupérer l'entier **N** saisi satisfaisant les conditions décrites précédemment, puis d'exploiter la fonction **GainPerd** afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface **" InterCHARIOT".**

NB : L'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution

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 **InterCHARIOT**	**3 pts**
Création et enregistrement du programme **PgmeCHAR**	**1 pt**
Développement de la fonction **GainPerd**	**6 pts**
Ajout des instructions : * De l'interface **InterCHARIOT** * Du module **Play**	**1 pt** **3 pts**
Importation des bibliothèques nécessaires, modularité et cohérence	**3 pts**
Implémentation de la procédure **REMP_AFFICHE** et son appel	**3 pts**

Lycée Béja Nord	**Sujet Pratique n°2**	Durée : 1 Heure
Classes : 4ème SC&M	Matière : Informatique — Enseignant : Kais TABOUBI	Date : **-**-2023

Important

1. *Une solution modulaire au problème posé est exigée.*
2. *Dans le répertoire **Bac2023**, créez un dossier de travail ayant comme nom votre numéro d'inscription (**6chiffres**) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.*

Une technique ultérieure de cryptographie consista à opérer non avec un décalage systématique, mais par une substitution aléatoire. Pour cela, on utilise un alphabet-clé, dans lequel les lettres se succèdent de manière désordonnée, par exemple : **HYLUJPVREAKBNDOFSQZCWMGITX**

C'est cette clé qui va servir ensuite à coder le message. Selon notre exemple, les **A** deviendront des **H**, les **B** des **Y**, les **C** des **L**, etc.

On se propose de saisir un message non nul contenant uniquement des lettres ***alphabétiques majuscules*** et des ***espaces*** puis d'effectuer et d'afficher son cryptage (l'alphabet-clé sera générer aléatoirement).

Exemple :

*Soit le tableau ci-dessous représente les **ordres des lettres alphabétiques majuscules** :*

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

❶ Pour le message **MES** et **l'alphabet-clé** suivants:

MES = "LE CRYPTAGE DES INFORMATIONS"

L'alphabet-clé :

W	R	L	J	G	T	D	P	K	Z	M	X	C	F	S	H	Y	B	O	A	E	U	V	Q	I	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

*Le programme affiche "***XG LBIHAWDG JGO KFTSBCWAKSFO***".*

- La lettre "**L**" sera cryptée par "**X**". **En effet**, l'ordre de la lettre "**L**" est égal à **11** dans l'ensemble des lettres alphabétiques, alors cette lettre sera cryptée par son équivalent dans **l'alphabet-clé** qui est la lettre "**X**".
- La lettre "**E**" sera cryptée par "**G**". **En effet**, l'ordre de la lettre "**E**" est égal à **4** dans l'ensemble des lettres alphabétiques, alors cette lettre sera cryptée par son équivalent dans **l'alphabet-clé** qui est la lettre "**G**".
- **Et ainsi de suite,…**

Ci-après, la déclaration **incomplète** de la fonction **Generer** à exploiter pour résoudre le problème posé, qui permet de générer aléatoirement une chaîne contenant les **26** lettres alphabétiques majuscules.

```
1  from random import randint
2  def generer():
3      ch=""
4      for i in range(26):
5          n=randint(65,90)
6          while ..................... :
7              n=randint(65,90)
8          ch+= .......
9      return ch
```

Question :

Pour compléter les pointillés il suffit de répondre aux tâches suivantes :

❶ **La ligne n°6 :** On répète la génération aléatoire du nombre tant que le caractère équivalent à **n** existe dans la chaine **ch.**

❷ **La ligne n°8 :** La concaténation de la chaine **ch** avec le caractère équivalent à **n.**

On se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : **"Cryptographie avec l'alphabet-Clé"**
- Un label demandant la saisie du message :
  "Introduire un message dont sa longueur appartient à l'intervalle [1, 100] \n et contient que des lettres majuscules et des espaces :"
- Une zone de saisie permettant la saisie du message.
- Un bouton intitulé **" Crypter"**
- Un bouton intitulé **"Effacer"** *(permettant d'effacer la zone de saisie et les deux labels d'affichage)*
- Un label permettant d'afficher la **clé** (résultat retourner par la fonction **Generer()**)
- Un label pour afficher le message adéquat.

Travail demandé :

1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom **"InterCRYPTE_AC"**.

2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom **"PgmeCRYPTE_AC ".**

3) Implémenter la fonction **Generer** tout en complétant les pointillés en répondant sur la question posée précédemment.

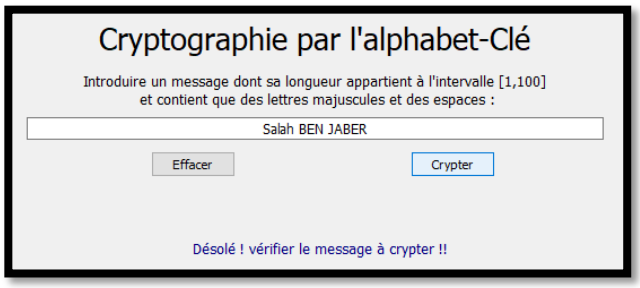
4) Développer dans le programme **" PgmeCRYPTE_AC "**, une fonction **Crypter (m, cle)** qui permet de crypter le message **m** en utilisant **cle** (résultat retourner par la fonction **Generer()**).

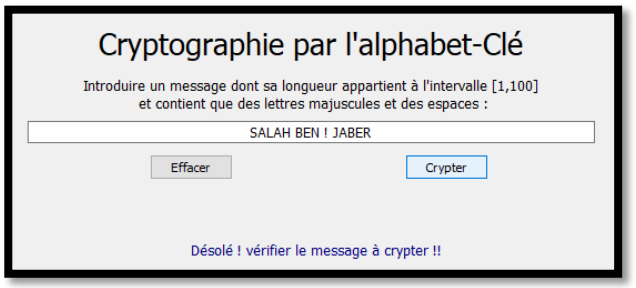
5) Dans le programme **" PgmeCRYPTE_AC "** :

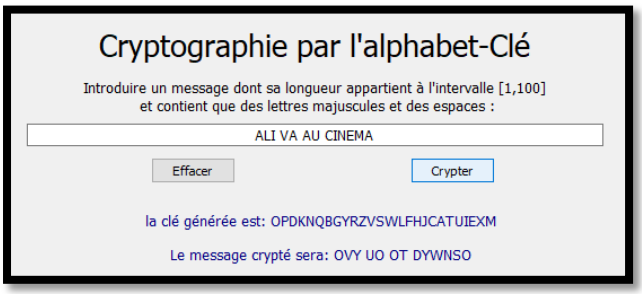
   - Ajouter les instructions permettant d'appeler l'interface graphique intitulée **" InterCRYPTE_AC "** en exploitant **l'annexe** ci-après.
   - Développer un module **"Play",** qui s'exécute suite à un clic sur le bouton **"Crypter",** permettant de récupérer le message **m** saisi satisfaisant les conditions décrites précédemment, puis d'exploiter la fonction **Crypter** afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface **" InterCRYPTE_AC ".**

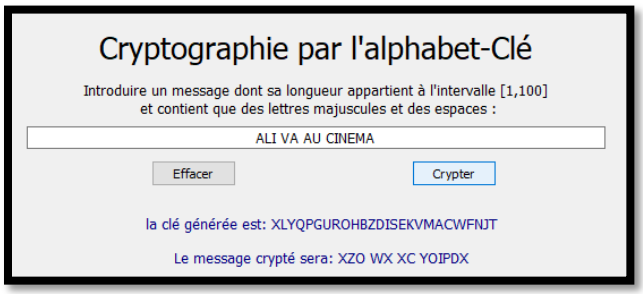
NB : L'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution









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 **InterCRYPTE_AC**	**3 pts**
Création et enregistrement du programme **PgmeCRYPTE_AC**	**1 pt**
Développement de la fonction **Crypter**	**6 pts**
Ajout des instructions : * De l'interface **InterCRYPTE_AC** * Du module **Play**	**1 pt** **3 pts**
Importation des bibliothèques nécessaires, modularité et cohérence	**3 pts**
Implémentation de la fonction **Generer** en complétant les pointillés	**3 pts**

Lycée Béja Nord	**Sujet Pratique n°3**	**Durée** : 1 Heure
Classes : 4ème SC&M	**Matière : Informatique** — **Enseignant : Kais TABOUBI**	**Date : **-**-2023**

Important

1. *Une solution modulaire au problème posé est exigée.*
2. *Dans le répertoire **Bac2023**, créez un dossier de travail ayant comme nom votre numéro d'inscription (**6chiffres**) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.*

On veut crypter une chaine de caractères données **CH** dont la taille ne dépasse pas **50** caractères en une chaine résultat **CHC** de la manière suivante :

- Parcourir la chaine **CH** de gauche à droite en comptant le nombre **d'occurrences successives** de chaque caractère de la chaine **CH**,
- Puis de ranger la chaine **CHC**, ce nombre suivi du caractère en question.

On se propose de saisir une chaîne de caractères **CH** qui doit être **non vide** et formée uniquement par ***des lettres alphabétiques,*** puis de former et d'afficher la chaine **CHC** selon le principe décrit précédemment.

Exemple :

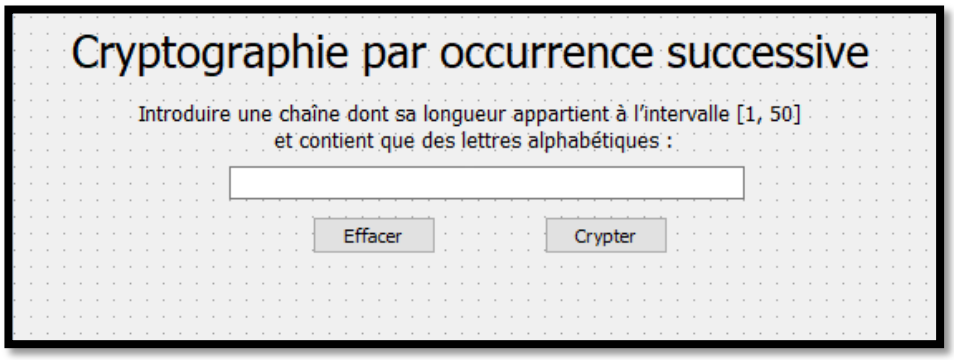
❶ Pour la chaine de caractères **CH** = "**aaaFyBsssssssssssazz**"

Le programme affiche **CHC :** "**3a1F1y1B12s1a2z**".

- **En effet,** on a **3** occurrences successives de la lettre "**a**", d'où la première partie de **CHC** contient "**3a**"
- **Ensuite,** on a **une seule** occurrence de la lettre "**F**", d'où la chaîne **CHC** devient "**3a1F**"
- **Et ainsi de suite,…**

On se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : **"Cryptographie avec occurrence successive"**
- Un label demandant la saisie de la chaîne :
  "Introduire une chaîne dont sa longueur appartient à l'intervalle [1, 50] \n et contient que des lettres alphabétiques :"
- Une zone de saisie permettant la saisie de la chaîne.
- Un bouton intitulé **" Crypter"**
- Un bouton intitulé **"Effacer"** *(permettant d'effacer la zone de saisie et le label d'affichage)*
- Un label pour afficher le message adéquat.



Travail demandé :

1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterCRYPTE_OS**".

2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**PgmeCRYPTE_OS ".**

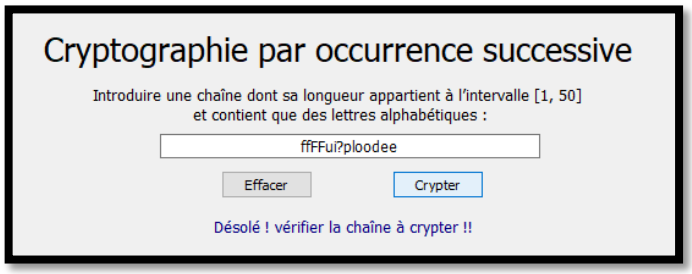
3) Développer dans le programme " **PgmeCRYPTE_OS** ", une fonction **Crypter** (**CH**) qui permet de crypter la chaîne **CH** en utilisant **le principe décrit précédemment**.

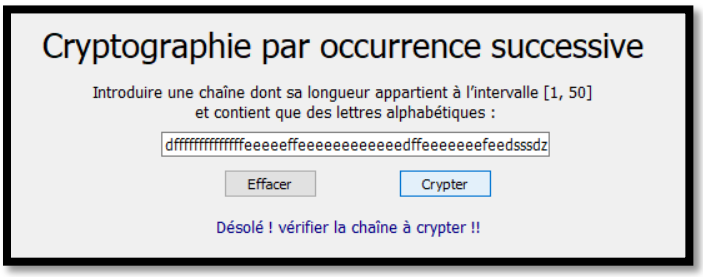
4) Dans le programme " **PgmeCRYPTE_OS** " :

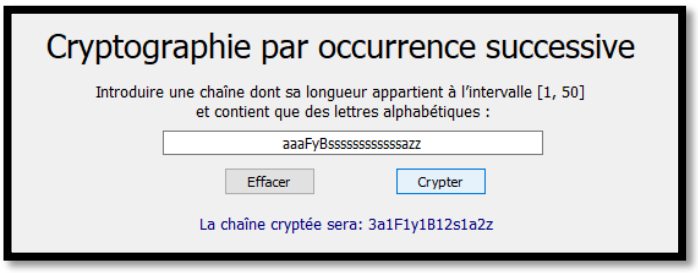
   - Ajouter les instructions permettant d'appeler l'interface graphique intitulée " **InterCRYPTE_OS** " en exploitant **l'annexe** ci-après.
   - Développer un module "**Play",** qui s'exécute suite à un clic sur le bouton "**Crypter",** permettant de récupérer la chaîne **CH** saisi satisfaisant les conditions décrites précédemment, puis d'exploiter la fonction **Crypter** afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface " **InterCRYPTE_OS".**

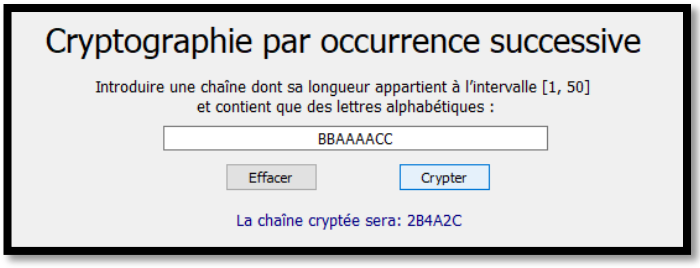
NB : L'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution









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 **InterCRYPTE_OS**	**3 pts**
Création et enregistrement du programme **PgmeCRYPTE_OS**	**1 pt**
Développement de la fonction **Crypter**	**6 pts**
Ajout des instructions : * De l'interface **InterCRYPTE_OS** * Du module **Play**	**2 pt** **5 pts**
Importation des bibliothèques nécessaires, modularité et cohérence	**3 pts**

Lycée Béja Nord	**Sujet Pratique n°4**	**Durée** : 1 Heure
Classes : 4ème SC&M	**Matière : Informatique** — **Enseignant : Kais TABOUBI**	**Date : **-**-2023**

Important

1. *Une solution modulaire au problème posé est exigée.*
2. *Dans le répertoire **Bac2023**, créez un dossier de travail ayant comme nom votre numéro d'inscription (**6chiffres**) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.*

On veut crypter un mot saisi **MS** non vide, composé uniquement par des lettres majuscules et d'afficher le mot crypté **MC**.

La méthode de cryptage est la suivante :

- Pour chaque lettre, déterminer son nombre d'occurrence (apparition) **n** dans le mot **MS**.
- Déterminer **K** qui est égal à **2*n** si **n** est impair et sera égal à **(n DIV 2)** si **n** est pair.
- Remplacer chaque lettre par **K^ième^** lettre qui la suit dans l'intervalle de l'alphabet [**'A'..'Z'**]. Pour les dernières lettres, on rend dès le début, par exemple si **K=3**, on remplacera **'A'** par **'D'**, **'B'** par **'E'**, **'C'** par **'F'**… **'Y'** par **'B'** et **'Z'** par **'C'**.

Exemple :

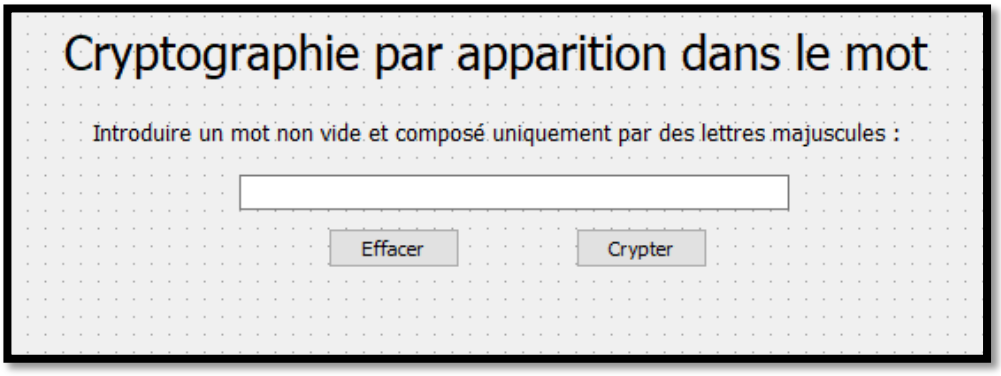
❶ Pour le mot **MS =** "**ZEBRE**"

	"Z"	**"E"**	**"B"**	**"R"**	**"E"**
Nombre d'apparition (n)	1	2	1	1	2
La valeur de K	2*1 = 2	2 **DIV** 2 = 1	2*1 = 2	2*1 = 2	2 **DIV** 2 = 1
La lettre de remplacement	**"B"**	**"F"**	**"D"**	**"T"**	**"F"**

Le programme affiche **MC :** "**BFDTF**".

On se propose de concevoir une interface graphique contenant les éléments suivants :

- Un label contenant le texte : **"Cryptographie par apparition dans le mot"**
- Un label demandant la saisie du mot:
  "Introduire un mot non vide et composé uniquement par des lettres majuscules :"
- Une zone de saisie permettant la saisie du mot.
- Un bouton intitulé **" Crypter"**
- Un bouton intitulé **"Effacer"** *(permettant d'effacer la zone de saisie et le label d'affichage)*
- Un label pour afficher le message adéquat.

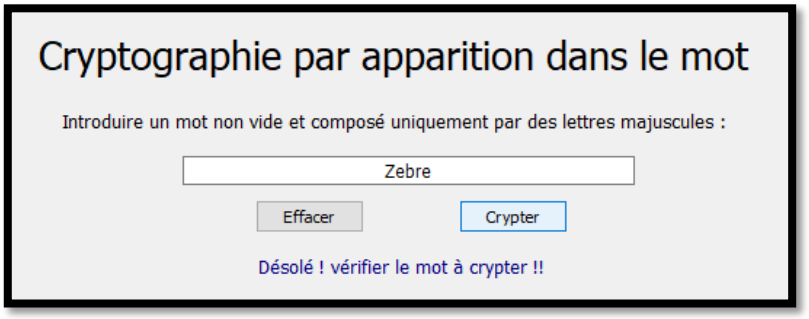


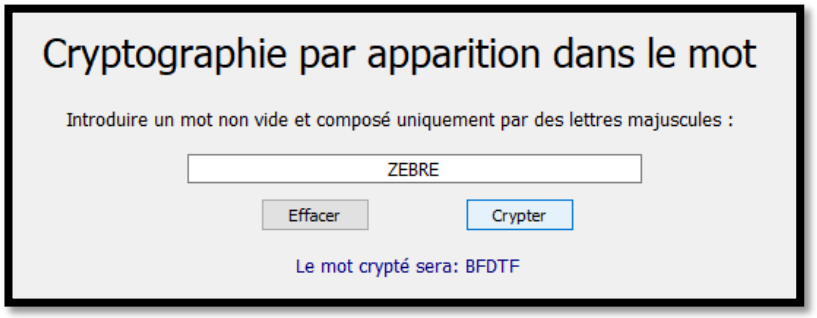
Travail demandé :

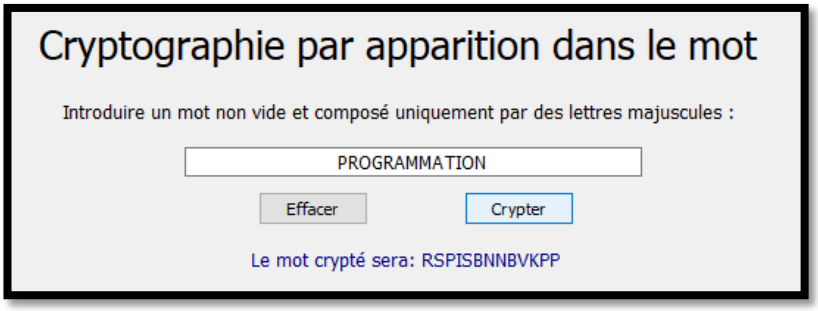
1) Concevoir une interface graphique comme illustrée ci-dessus et l'enregistrer, dans votre dossier de travail, sous le nom "**InterCRYPTE_AM**".
2) Créer un programme Python et l'enregistrer, dans votre dossier de travail, sous le nom "**PgmeCRYPTE_AM ".**
3) Développer dans le programme " **PgmeCRYPTE_AM** ", une fonction **Crypter (MS)** qui permet de crypter le mot **MS** en utilisant **le principe décrit précédemment**.
4) Dans le programme " **PgmeCRYPTE_AM** " :
   - Ajouter les instructions permettant d'appeler l'interface graphique intitulée " **InterCRYPTE_AM** " en exploitant **l'annexe** ci-après.
   - Développer un module **"Play",** qui s'exécute suite à un clic sur le bouton **"Crypter",** permettant de récupérer le mot **MS** saisi satisfaisant les conditions décrites précédemment, puis d'exploiter la fonction **Crypter** afin d'afficher le message adéquat via le **label** dédié à l'affichage de l'interface " **InterCRYPTE_AM ".**

NB : L'affichage du message doit être conforme aux exemples d'exécution suivants :

Exemples d'exécution







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 **InterCRYPTE_AM**	**3 pts**
Création et enregistrement du programme **PgmeCRYPTE_AM**	**1 pt**
Développement de la fonction **Crypter**	**6 pts**
Ajout des instructions : * De l'interface **InterCRYPTE_AM** * Du module **Play**	**2 pt** **5 pts**
Importation des bibliothèques nécessaires, modularité et cohérence	**3 pts**