

## Création et exécution d'un QCM

L'idée générale est de répondre à une même problématique par programmation par fonctions en utilisant des types différents. La problématique proposée est la construction et l'exécution d'un QCM.

On pourra répondre à cette problématique en utilisant :

- des chaînes de caractères
- des listes
- un dictionnaire
- en faisant appel à un fichier extérieur (texte ou tableur)
- une page web

Cette problématique peut être un exercice récurrent tout au long de l'année pour mettre en application les différents types étudiés, peut servir de mini projet ou un TP finalisant une séquence.

L'activité présentée ci-après est sous forme de TP et cherche à répondre à la question en utilisant les chaînes de caractères.

Intérêt : cela peut être fait très tôt dans l'année et permet aux élèves de prendre de bonnes habitudes de programmation (utilisation de fonctions avec paramètres, spécifications des fonctions)

Les pré-requis : savoir manipuler print, input, chaîne de caractères, la notion de fonction, boucle et conditionnelles

Certains exercices manipulant les chaînes et les fonctions pourront avoir été travaillés préalablement.  
Inverse(Txt) qui prend en entrée une chaîne de caractères et renvoie la même chaîne écrite à l'envers  
Inverse(" Ma chaîne ") → " eniahc aM "  
Insert (Txt,c) insère le caractère c entre chaque lettre de Txt : Insert(" NSI ", " . ") --> " N.S.I "  
Supprime(Txt,c), CompteCar(Txt,c), EstPresent(Txt,c) ...

Programmation : Les chaînes de caractères en Python, lecture, construction et affichage.  
Fonctions

Elèves : 1<sup>ère</sup> NSI ou Terminale ISN

Passation : En groupe sur deux séances. Seul ou en binôme et en autonomie.

Lien avec le programme (NSI) : Langages et programmation

Constructions élémentaires	Mettre en évidence un corpus de constructions élémentaires.
Spécification	Prototyper une fonction. Décrire les préconditions sur les arguments. Décrire des postconditions sur les résultats.
Mise au point de programmes	Utiliser des jeux de tests.

### Présentation de la suite du document

Les 4 pages suivantes sont la présentation du TP version élève.

Une première page qui présente la problématique, les pages suivantes le travail à effectuer.

Afin de mieux contrôler le travail fait par l'élève ou le binôme, une phase de validation doit être effectuée par l'enseignement, une partie commentaire peut aussi être utilisée (aide donnée, problème(s) rencontré(s), réussite, prise d'initiative ... ce qui permet de faire aussi de ce TP une activité qui peut être évaluée.

Les premières tâches sont guidées (l'idée n'étant pas l'algorithme mais le codage en Python), les algorithmes sont donnés, restent (et ce n'est pas forcément toujours simple) à coder en Python et faire des jeux d'essai.

Une fois la phase de création du QCM réalisée, on travaille avec une chaîne constante ce qui évite à chaque fois de « reconstruire » le QCM. A ce stade devrait émerger le problème du stockage et pour les plus avancés ou pour tous, une écriture, puis lecture dans un fichier texte peut enrichir le TP.

La fin, consistant à l'exécution du QCM est nettement moins guidée.

### Les questions à priori / prolongement / critique

Comment stocker un QCM créé ?

Enregistrement de données dans un fichier texte.

Que se passe-t-il si on insère le caractère « saut de ligne » ?

Même TP avec des listes et fichiers csv

Même TP avec web (voir fiche activité web) ...

## Présentation du TP

On souhaite construire et exécuter un QCM de plusieurs questions.  
Chaque question propose un énoncé et trois réponses dont une seule est juste, par exemple :

$2^{10} =$       A : 512      B : 1024      C : 2048      (La bonne réponse est B)

Le TP propose une façon de répondre à la question en n'utilisant que des chaînes de caractères.  
Une question sera une chaîne dont les différents éléments (énoncé, réponse A, réponse B, réponse C et lettre de la bonne réponse) seront séparés par un caractère spécial qui sera dans un premier temps le point virgule.

Par exemple la question ci-contre sera représentée par la chaîne de caractères:  
 $2^{10} =$ ; A : 512; B : 1024; C : 2048;B

2<sup>10</sup> =  
A : 512  
B : 1024  
C : 2048

Le QCM sera une chaîne de questions séparées par un autre caractère spécial qui sera dans un premier temps le dollar.

Le QCM ci-contre constitué de deux questions sera représenté par la chaîne de caractères

$2^{10} =$ ; A : 512; B : 1024; C : 2048;\$4\*7=; A : 24; B : 26; C : 28;C\$

Question n°1  
2<sup>10</sup> =  
A : 512  
B : 1024  
C : 2048  
Question n°2  
4\*7=  
A : 24  
B : 26  
C : 28

A la fin du TP, on devrait pouvoir obtenir une exécution du type :

```
2^10 =
A : 512
B : 1024
C : 2048
-----
Votre reponse : B
9*7=
A : 56
B : 61
C : 63
-----
Votre reponse : C
11 en decimale s'ecrit en binaire
A : 1001
B : 1011
C : 1101
-----
Votre reponse : A
Votre score est : 2 bonnes reponses
```

### Attention

Toutes les fonctions seront correctement spécifiées  
On organisera correctement les fonctions  
On commentera si nécessaire le codage

## Enoncé du TP

**Bien lire chacune des questions dans leur totalité avant de les exécuter, afin de bien comprendre ce qui est demandé.**

### 1) Création et affichage d'une question

a) Coder en Python la fonction **CreQuestion()** dont l'algorithme est le suivant :

#### Fonction CreQuestion()

```
# Pas d'entrée
# Sortie une chaîne de caractères
Initialiser la variable Txt à vide
Lire l'énoncé de la question et ajouter à Txt le résultat ainsi que le caractère point virgule
Lire la première réponse proposée, ajouter à Txt la chaîne « A : », la réponse et le caractère ;
Lire la deuxième réponse proposée, ajouter à Txt la chaîne « B : », la réponse et le caractère ;
Lire la troisième réponse proposée, ajouter à Txt la chaîne « C : », la réponse et le caractère ;
Lire la bonne réponse (A, B ou C), ajouter à Txt la bonne réponse.
Retourner la chaîne Txt
```

b) Coder en Python la fonction **AffQuestion(Q)** dont l'algorithme est le suivant :

#### Fonction AffQuestion(Q)

```
# Entrée : Une chaîne de caractères Q (représentant une question)
# Sortie : Un affichage
Initialiser une variable chaîne Txt à vide
Lire caractère par caractère la chaîne Q sauf le dernier
Si le caractère lu est un point virgule
    Afficher Txt
    Initialiser Txt à vide
Sinon
    Ajouter à Txt le caractère lu
Afficher une ligne de 15 tirets # tiret du 6
```

c) Exécuter le code suivant :

```
MaQuestion=CreQuestion()
print(MaQuestion)
AffQuestion(MaQuestion)
```

et modifier si nécessaire les fonctions si le rendu ne vous satisfait pas

d) Jeu d'essai possible :

```
>>>
Enonce : 2^10 =
Reponse A : 512
Reponse B : 1024
Reponse C : 2048
Bonne reponse : B
2^10 =; A : 512; B : 1024; C : 2048;B
2^10 =
A : 512
B : 1024
C : 2048
-----
```

```
MaQuestion = CreQuestion()
Lecture de l'énoncé, des réponses et de
la lettre de la bonne réponse
```

```
print(MaQuestion)
La question sous la forme d'une chaîne de caractère
```

```
AffQuestion(MaQuestion)
Affichage de la question comme le verra l'utilisateur
```

Appeler l'enseignant pour validation ou en cas de problème

Commentaires :

## 2) Création et affichage d'un QCM

a) Coder la fonction **CreQCM()** dont l'algorithme est le suivant :

### Fonction CreQCM(n)

```
# Entrée : un entier naturel n (nombre de questions du QCM)
# Sortie une chaîne de caractères
Initialiser la variable Txt à vide
Pour k allant de 0 à n-1
    Afficher le texte « Question n° » suivi du numéro de la question à lire
    Ajouter à Txt une question
    Ajouter à Txt le caractère $
```

Remarque : on utilisera la fonction CreQuestion() écrite précédemment.

b) Compléter l'algorithme suivant permettant d'afficher un QCM, puis coder cette fonction en Python.

### Fonction AffQCM(qcm)

```
# Entrée : Une chaîne de caractères qcm (représentant un QCM)
# Sortie : Un affichage
Initialiser une variable chaîne Txt à vide
Lire caractère par caractère la chaîne qcm
    Si le caractère lu est un dollar
        .....
        Initialiser Txt à vide
    Sinon
        Ajouter à Txt le caractère lu
```

c) Exécuter le code suivant :

```
MonQCM=CreQCM(2)
print(MonQCM)
AffQCM(MonQCM)
```

et modifier si nécessaire les fonctions si le rendu ne vous satisfait pas

d) Jeu d'essai possible :

```
>>>
Question n° 1
Enonce : 2^10 =
Reponse A : 512
Reponse B : 1024
Reponse C : 2048
Bonne reponse : B
Question n° 2
Enonce : 9*7 =
Reponse A : 56
Reponse B : 61
Reponse C : 63
Bonne reponse : C
2^10 =; A : 512; B : 1024; C : 2048;B$9*7 =; A : 56; B : 61; C : 63;C$
2^10 =
A : 512
B : 1024
C : 2048
-----
9*7 =
A : 56
B : 61
C : 63
-----
```

Appeler l'enseignant pour validation ou en cas de problème

Commentaires :

### 3) Modification des fonctions précédentes

On aimerait laisser plus de liberté et que les éléments d'une question soit séparés par un caractère (ou chaîne de caractères)  $c_1$  et les questions du QCM par un caractère (ou chaîne de caractères)  $c_2$ .

Apporter les modifications nécessaires aux fonctions écrites précédemment et faire un jeu d'essai d'affichage du QCM avec la chaîne MonQcm qui sera construite de la façon suivante (on n'utilise pas la fonction CreQcm(...)) :

On pourra copier les fonctions et modifier les copies (il faut toujours garder ce qui fonctionne !!)

Q1="2^10 =\$ A : 512\$ B : 1024\$ C : 2048\$B" # le séparateur de question est \$

Q2="9\*7=\$ A : 56\$ B : 61\$ C : 63\$C;"

Q3 = A vous de l'inventer

MonQcm=Q1+"&"+Q2+"&"+Q3+"&" # le séparateur de question est &

Appeler l'enseignant pour validation ou en cas de problème

Commentaires :

### 4) Listes des bonnes réponses

a) Coder la fonction **ListBonRep(qcm, c2)** dont l'algorithme est le suivant :

#### Fonction ListBonRep(qcm,c2)

# Entrée : Une chaîne de caractères qcm (représentant un QCM) et c2 un caractère

# Sortie : Une chaîne de caractères (qui est la liste des lettres des bonnes réponses)

Initialiser une variable chaîne Txt à vide

Lire caractère par caractère la chaîne qcm

Si le caractère lu est le caractère c2

Ajouter à Txt le caractère lu précédemment

Retourner la chaîne Txt

b) Exécuter le code suivant :

```
c1="$"  
c2="&"  
BR=ListBonRep(MonQcm, c2)  
print(BR)
```

### 5) Exécution du QCM

a) Ecrire la fonction **LireRep()** qui demande à l'utilisateur sa réponse et renvoie sa réponse sous la forme de caractère. (très simple, nécessite que très peu d'instruction(s))

b) En s'inspirant de la fonction AffichQCM(...) écrire une fonction **ExeQCM(arguments nécessaires)** qui affiche le QCM, lit la réponse de l'utilisateur à chaque question et renvoie les réponses données sous la forme d'une chaîne de caractères.

c) Ecrire la fonction **Score(BR,Rep)** dont les arguments sont les chaînes de caractères BR (les bonnes réponses) et Rep (la chaînes de caractères correspondant aux réponses de l'utilisateur) et renvoie un entier égal au nombre de bonnes réponses.

d) Ecrire un programme principal exécutant le QCM et affichant le nombre de bonnes réponses.

Appeler l'enseignant pour validation ou en cas de problème

Commentaires :