



HAI405I

LE SUJET DÉVOILÉ

16 JANVIER 2023



Les groupes ont été complétés pour être de 4.

Si vous n'êtes dans aucun groupe, contacter les deux responsables rapidement.

Il existe de nombreux sites qui proposent des exercices de mathématiques :

- soit sous forme de qcm pour que les étudiants s'entraînent ou que les enseignants vérifient la connaissance du cours,
- soit sous forme d'une base d'exercices à destinations des enseignants.

Mais il existe peu de ressources du genre pour des exercices en informatique.


On souhaite donc développer une application web où des enseignants pourraient s'identifier et proposer des énoncés d'exercices. On se limitera à des questions sous forme de QCM.


L'utilisateur peut créer des *questions*, ainsi que modifier ou supprimer des questions existantes. Chaque question comporte un *énoncé*, des propositions de *réponses* ainsi que des *étiquettes* thématiques.


Pour aider l'utilisateur à saisir/éditer ses questions, l'application doit comporter un *éditeur* pour la saisie et un *visualisateur* pour afficher le rendu.

L'enseignant peut ensuite, à partir de sa base de questions et des étiquettes, générer une page de questions.

Question 1
L'arbre suivant
``mermaid
Graph TB
A --- B
A --- C
...
est

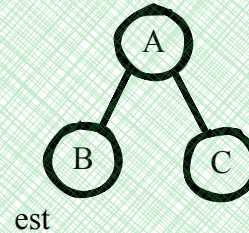
☐ de profondeur 2 

☐ parfait 

☐ équilibré 

Aperçu

Question 1
L'arbre suivant



☐ de profondeur 2

☐ parfait

☐ équilibré

Retour

L'application doit permettre la création de comptes utilisateur.

Chaque utilisateur, une fois identifié, peut créer de nouveaux énoncés ou modifier des énoncés dont il est le créateur.

Chaque question peut comporter un nombre quelconque d'étiquettes. Par exemple, une question peut être étiquetée à la fois par *graphe*, *algorithmique*, *java*.

Pour chaque question, l'utilisateur choisit parmi les étiquettes déjà existantes mais peut aussi en ajouter des nouvelles.

Ce contenu dans l'éditeur

```
## titre H2
### sous titre H3
**en gras**, *en italique*...
- un item
- un second item
```

donnera dans la partie visualisation

titre H2

sous titre H3

en gras, *en italique*...

- un item
- un second item

L'énoncé doit pouvoir contenir des éléments mathématiques écrits en *LaTeX* encadrés par `$$` ou `$`. Par exemple, on doit pouvoir écrire `$e^{i\pi}+1=0$` qui s'affichera comme $e^{i\pi} + 1 = 0$ ou bien `$$e^{i\pi}+1=0$$` qui donnera

$$e^{i\pi} + 1 = 0$$

Des bibliothèques existent et vous simplifieront la vie.

```
```python
def fact(n):
 if n<2:
 return ...
 else:
 return ...
```
```

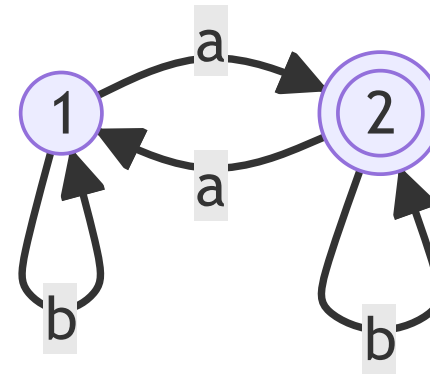
devrait s'afficher dans la visualisation comme ceci

```
def fact(n):
    if n<2:
        return ...
    else:
        return ...
```

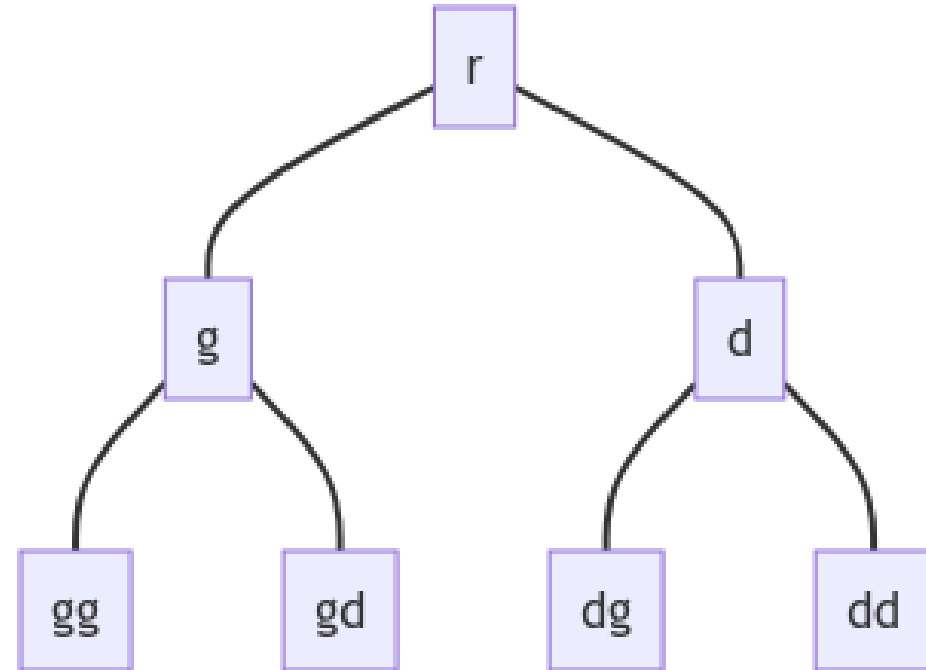
Des bibliothèques existent et vous simplifieront la vie.

Il doit être possible de décrire textuellement n'importe où dans l'énoncé des graphiques de différentes sortes (graphes, arbres, automates, diagrammes UML...). Il vous est suggéré de regarder les bibliothèques **mermaid** ou **kroki**.

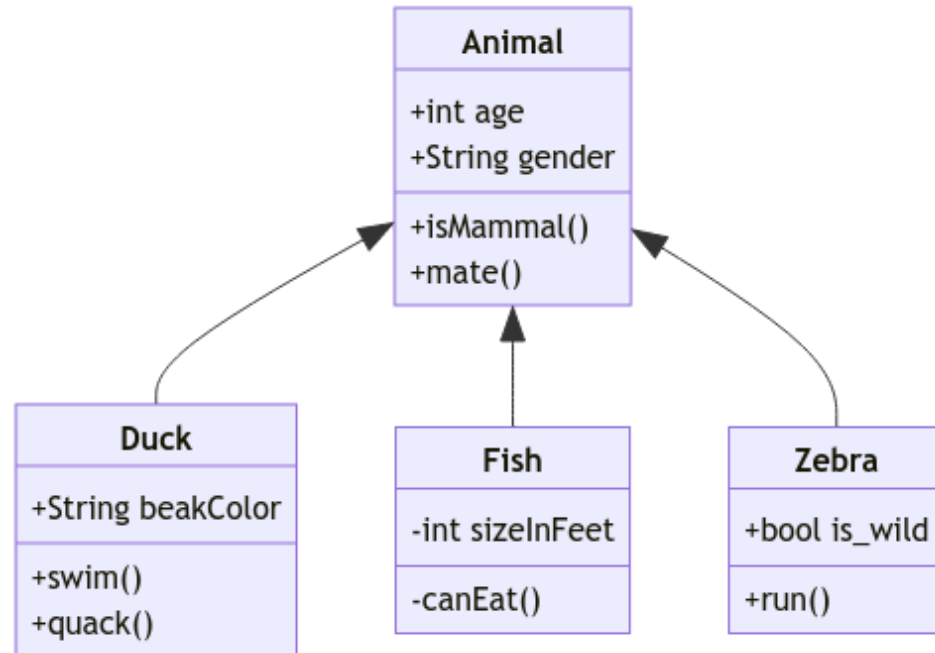
```
```mermaid
graph LR
 A((1)) -->|a| B(((2)))
 B -->|a| A
 B -->|b| B
 A -->|b| A
```
```



```
```mermaid
graph TB
 r --- g
 r --- d
 g --- gg
 g --- gd
 d --- dg
 d --- dd
```
```



```
```mermaid
classDiagram
 Animal <|-- Duck
 Animal <|-- Fish
 Animal <|-- Zebra
 Animal : +int age
 Animal : +String gender
 Animal: +isMammal()
 Animal: +mate()
 class Duck{
 +String beakColor
 +swim()
 +quack()
 }
 class Fish{
 -int sizeInFeet
 -canEat()
 }
 class Zebra{
 +bool is_wild
 +run()
 }
```
```



Pour chaque énoncé, l'utilisateur saisit les différentes propositions de réponses en indiquant si elles sont correctes ou non.

Idéalement, il faut permettre du latex, des graphiques et du code dans les propositions de réponse comme cela a été fait pour les énoncés.

GÉNÉRATION DE PAGE DE QUESTIONS



Un enseignant identifié peut, parmi ses énoncés, choisir un certain nombre d'exercices pour générer une page en y ajoutant un titre dans l'optique de pouvoir imprimer des feuilles d'exercices.

Il doit être possible de filtrer par étiquette pour faciliter la conception de la page.



CONTRAINTES TECHNIQUES



L'application est une application web dont le serveur est en **python** et utilise **Flask**.

Coté client, évidemment **HTML**, **CSS**, **JS** sont autorisés ainsi que toute bibliothèque web.

Les énoncés (incluant donc la question, les propositions de réponses ainsi que la correction) doivent être enregistrés sur le serveur dans un fichier texte et dans un format à définir.

Réfléchissez bien à votre format. Il doit être facilement possible de l'adapter à d'autres types de questions (Vrai/Faux, numériques...)

Les étiquettes seront également stockées dans un fichier texte côté serveur.



La démonstration a lieu le lundi 23 janvier à partir de 13h15. Tous les membres doivent être présents.

Elle durera environ 15 minutes incluant 5 minutes de démonstration (sans diaporama) proprement dite (qui devra donc être préparée) et des questions individuelles ou de groupe.

La démonstration peut avoir lieu sur un ordinateur de la fds ou sur une machine personnelle.

La date limite pour le rendu est le 23 janvier à 13h15. Mais il est possible de rendre dès le vendredi 20.

Le rendu est constitué d'un unique fichier sous forme d'archive contenant :

- un README avec les explications concernant l'installation et le lancement de l'application ainsi que la description du format de sauvegarde des énoncés,
- l'ensemble des fichiers sources,
- une **Webographie** indiquant toutes les bibliothèques utilisées ainsi que **toutes** vos sources d'inspiration,
- la liste de tout ce qui ne fonctionne pas ou qui n'est pas fait.

- Il ne s'agit pas de tout re-développer, pensez bibliothèques.
- Les priorités de développement sont :
 - i. Structure de base où les utilisateurs peuvent créer/modifier un énoncé (question et réponses) en texte brut y compris la sauvegarde sur le serveur,
 - ii. Inclure le **Markdown** et sa visualisation,
 - iii. Identification des utilisateurs,
 - iv. Ajouter les suppléments au Markdown (Coloration de code, LateX, graphiques) et leurs visualisations,
 - v. Gérer les étiquettes,
 - vi. Génération de pages de questions.

REMARQUES, ENCORE

- Produire du code lisible, compréhensible, commenté et testé,
- Prendre soin de l'ergonomie de l'application.
- 4 salles disponibles : 36.204, 36.206, 36.207 et 36.208.
- Chaque groupe se place ensemble dans la salle de son choix.

Rappel, le plagiat est illégal et donc sanctionné.

Si vous avez adapté de larges portions de codes trouvés en ligne

- l'ajouter dans la webographie,
- le signaler dans un commentaire dans le code.

AU COURS DE CETTE SEMAINE



Si il y a des éclaircissements à apporter sur le sujet durant la semaine, poser vos questions à l'enseignant dans la salle.

Les réponses seront apportées à tous lors de la demi-journée suivante ou plus tôt en cas de blocage urgent.

Signaler au plus tôt les problèmes dans les groupes.



QUESTIONS ?

Des questions ?

Go !!!

