Projet Programmation Web

Date limite de dépôt sur moodle : 3 mai 2020 à 23h
À réaliser en binôme

Mis à jour le 1er avril 2020

1 Contexte

Un *problème de Parson* est une forme d'exercice utilisée dans l'enseignement des bases de la programmation. Il consiste à déplacer des lignes de texte pour former un programme répondant à un problème donné. L'étudiant peut placer les lignes les unes au dessus des autres ainsi que de droite à gauche pour simuler l'indentation. La figure 1 présente un exemple de tel problème.

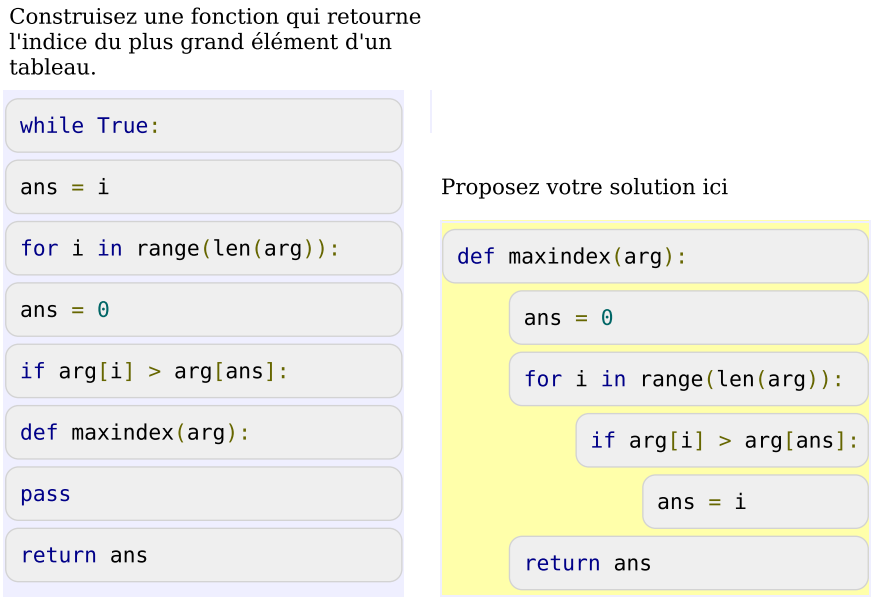

FIGURE 1 – Exemple de problème de Parson : énoncé (à gauche) et solution (à droite)

2 Instructions

Développez une application web, dont le backend sera **nécessairement** basé sur (au choix) :

— **php + symphony** ;
— ou une/des autres technologies web non encore abordées (par exemple, scala, kotlin, clojure, ocaml/ocsigen,...) après aprobation par email par les enseignants avant le 05 avril 2020 ; sont explicitement **exclus** : php, java, perl, C, python, nodejs ou autres approches directement basées sur javascript.

Son apparence **devra** être claire, structurée, et adaptée sur diverses tailles d'écran, du smartphone à l'ordinateur de bureau (pour cela, Bootstrap est *recommandé*). La base de donnée **devra** être bien structurée, *par exemple* en utilisant doctrine. On pourra aussi *par exemple* utiliser jquery pour dynamiser les pages. D'autres bibliothèques et frameworks [1] *pourront* être utilisées **en plus**, du moment qu'elles ne

1. mais pas de code simplement récupéré d'internet sans identification claire

rendent pas le projet trivial, et que leur utilité est explicitement justifiée. Il **faudra** fournir **au moins** les fonctionnalités suivantes :

- une page d'inscription permet de s'enregistrer (puis de s'authentifier) en tant qu'*enseignant* ou qu'*étudiant* ;
- un *enseignant* peut créer des *cours*, composés de listes d'*exercices* ;
- un *enseignant* peut créer des *exercices* de type *problème de Parson* en écrivant des instructions, fournissant une liste de lignes, et une solution construite à partir de ces lignes ;
- un *étudiant* peut s'inscrire à un *cours* ;
- une interface permet à un *étudiant* de tenter de résoudre un *exercice* et qui valide [2] ou non la réponse de manière **asynchrone** (i.e. utilisation de l'approche AJAX **requise**) ;
- une interface permet à un *étudiant* de visualiser les exercices réussis ;
- une interface permet à un *enseignant* de visualiser le taux de réussite et le nombre de tentatives de ses *étudiants* pour chaque exercice.

3 Évaluation

Le projet est à réaliser en binôme, et s'achèvera par une soutenance de 20 minutes (10 minutes de présentation, 10 minutes de questions). Les deux membres du projet devront pouvoir s'exprimer sur l'intégralité du projet réalisé. La soutenance consistera à présenter une démonstration en direct du site web, ainsi qu'à décrire une sélection d'éléments techniques intéressants de l'implémentation. On devra s'assurer à l'avance que le site web est accessible sur la machine utilisée pour la présentation. Celui-ci devrai être le même que celui déposé sur moodle à la date limite du dépôt.

La soutenance permettra notamment d'évaluer la qualité de la présentation et la capacité à répondre à des questions techniques.

La partie technique du projet sera notamment évalué selon les critères suivants :

Critère	minimal	moyen	optimal
Inscription	pas d'interface	inscription avec identifiant et authentification par mot de passe	inscription avec identifiant et authentification par mot de passe, rôles enseignant/étudiant bien séparés (pages enseignant inaccessibles aux étudiants)
Création de problème	pas d'interface	interface simple avec liste d'instructions prédéfinies	interface complète avec création d'instructions et possibilité d'assigner plusieurs solutions à un même exercice
Résolution de problème	pas d'interface	validation (correct/incorrect) par rechargement de la page	validation dynamique et feedback précis (surlignage des lignes/indentation incorrectes)
Résultats étudiant	pas d'interface		interface complète avec liens vers chaque exercice et statut de résolution
Résultats enseignant	pas d'interface		interface complète avec résultats par cours/exercice/étudiant

2. Il ne s'agit pas d'exécuter du code, mais simplement de comparer la solution à la réponse de l'étudiant