

Indications sur les exercices d'informatique à rendre

Veuillez m'excuser pour tous les mots mal orthographiés.

⚠ Disclaimer ⚠:

Les contenus de ce fichier viennent de l'interprétation personnelle de l'auteur sur lesdites exercices. Donc, certaines affirmations pourraient être biaisées voire fausses. Si vous continuez quand même de poursuivre votre lecture, vous acceptez le risque d'être induit en erreur et l'auteur n' en sera aucunement responsable.

exo 1:

Écrire un algorithme et un programme en Python qui permet d'afficher la table de vérité d'une fonction logique donnée, puis afficher la première forme et la seconde forme canonique de la fonction en question.

exo 2:

Écrire un algorithme et un programme en Python permettant de minimiser une fonction logique donnée par la méthode de Karnaugh.

Sur Python:

L'objectif des exercices c'est l'initiation des étudiants à la programmation. Python est un langage de programmation très utilisé dans le monde (en IA, Développement web & desktop, Jeux, ...) et réputé pour être facile à apprendre.

Comme il s'agit d'initiation, dans les exercices, on va programmer en procédural (et non en orienté objet (POO)).

Pourquoi en procédural?

Parce que c'est la manière primitive de programmer. (Je vous invite à en savoir plus sur ce qu'est la programmation procédurale).

Indications:

- Aucune classe ne doit être présente dans votre code puisque les classes c'est pour la POO. L'image suivante montre ce qu'est une classe en python.

```
class Setting(BaseModel):
    audiocodecs: str

class Section(BaseModel):
    settings: Setting
    account1: AccountSIP
```

- Vous ne devez utiliser aucun module ou librairie Python i.e votre code ne doit comporter des mots "import" situés au début de votre code. Les exercices sont réalisables sans aucun import que ce soit et je suis sûr que c'est l'objectif des exercices (nous avons déjà tout ce dont on a besoin (boucle, condition, fonction)). Je témoigne que je les ai réalisés dans ce condition.
- Les codes sources de chaque étudiant (dans le fichier .py) ne devraient jamais être identiques. La structure de code entre deux ou plusieurs étudiants doit être différente (même si vous décidez de changer les noms des variables, les boucles utilisées, etc... si la structure est la même, ça va se savoir).

Voici le lien du site officiel de Python: <https://www.python.org/>

Si vous êtes sur le point d'apprendre Python de zéro, tapez sur le moteur de recherche Google : "Get started with Python". Les résultats sont tous pertinents.

Sur l'algo:

Ici, l'objectif est de savoir écrire des algorithmes de manière rigoureuse. Je vous conseille de voir python avant les algorithmes pour un aperçu de manière très concrète sur ce que sont les variables, les conditions (si, sinon si, si) et les boucle (pour, tant que). Les initiés vont sûrement me contredire là-dessus.

Pour ma part, j'ai d'abord programmé les exercices sur python puis les traduire en syntaxe algorithmique. Le but étant de vérifier que mon programme fonctionne et le généralisé en algorithme. Je vous conseille de faire de même parce que c'est un gain de temps considérable de voir tout de suite que l'algorithme que j'ai élaboré fonctionne ou non. Ici "Écrire un algorithme et un programme Python", c'est l'inverse qui est le mieux.

Je reviens sur ce que j'ai dit dans "**Sur Python**", si vous avez par exemple utilisé des classes ou des modules/librairies python, comment allez vous les traduire en langage algorithmique ? (C'est vraiment une question que je vous pose 😊. si vous avez la réponse, svp dites le moi 🙏).

Sur GitHub:

En gros, il s'agit d'un espace où on dépose les codes sources pour en avoir une version unique et de collaborer avec d'autres développeurs.

(Je vais peut être créer une video pour la création et la modification des droits d'accès sur github)

Créez un fichier [readme.md](#) dans votre dépôt github pour expliquer à l'utilisateur de votre programme comment il fonctionne. Par exemple, sur l'utilisation des opérateurs: "OR" ou "+".

Sur les exercices:

- L'un des deux ou les deux à la fois ?
Je ne sais pas. Mais en raisonnant par pire cas, on doit faire les deux: Si l'un des deux alors envoyez l'un des deux; sinon, comme vous avez fini les deux, envoyez les. Faire les deux exercices est moins risqué que de n'en faire qu'un.

- Lequel des deux est le plus difficile ?
Je dirai l'exo 2 car pour réaliser la méthode de Karnaugh, on doit passer par la table vérité puis la table de Karnaugh.
La table de vérité étant dans l'exo 1, on doit donc passer par l'exo 1 pour l'exo 2. (là c'est vraiment mon avis, je rappelle ici le Disclaimer 😊).
- Combien de temps ça m' a pris de faire les exercices ?
Je ne les ai pas encore fini mais j'y suis à 60% environ (il est tout à fait possible que je ne puisse les finir à temps) et jusque là ça m' a pris 26h à peu près en réalisation continue.
- Alors finalement, les exercices sont difficiles ?
Je dirai Oui car, et c'est mon avis, les instructions des exercices données ne sont pas très explicites donc les possibilités de construire les algorithmes sont infinies. De plus, pour arriver là où j'en suis j'ai effectué pas mal de recherches (analyse des expressions algébriques, codage de Gray, méthode de Karnaugh, etc). Je suis sûr que je ne suis pas le seul d'ailleurs 😊.
- Pourquoi donner le nombre de variables selon l'instruction du professeur dans le programme ?
Cette question est vraiment très pertinente et je ne connais pas la réponse. Pourquoi ne pas juste entrer l'expression algébrique de la fonction logique puis de compter le nombre de variables dans ladite expression ?
- On doit aussi accompagner un fichier `algorithme.txt` avec chaque programme des deux exercices ?
Sur votre dépôt github, vous allez placer le programme `.py` et un fichier `algorithme.txt` contenant l'algorithme demandé dans chaque exercice. Vous pouvez aussi placer votre algorithme dans le fichier `readme.md` en spécifiant dans ce dernier que l'algorithme est là.
- Pourquoi rédiger ce fichier et faire tout ceci ?
J'en sais rien, pour aider... 😊

Si vous avez des questions ou des réponses de plus, on va en parler dans les groupes de discussions sur Messenger...

Je ne répondrai à aucun MP de votre part car vos remarques pourraient être utiles pour tout le monde donc visible dans le groupe de discussion...

Bien sûr, il y en a qui ont sûrement espéré trouver les solutions des exercices dans ce fichier alors désolé c'est une indication. 😊

Merci 🙏