

# ASYNCONF

## TOURNOI D'ALGORITHMIQUE

### Thème :

# Halloween



### Spécifications :

Les `[entrées]` sont des données variables, qui doivent être entrées via la console.

Les `{sorties}` sont des données qui doivent être affichées dans la console, dans le format défini par l'exercice.

Les principaux points qui seront évalués sont :

- Les fonctionnalités : le code doit marcher pour les tests donnés dans chaque exercice.
- La fiabilité : le code doit marcher pour des tests non donnés, mais conformes à la consigne.
- La robustesse : le code doit pouvoir marcher dans toutes les situations et prévoir les cas où des erreurs peuvent survenir.
- La lisibilité : le code doit être organisé, aéré et explicite.
- L'optimisation : le code doit effectuer l'exercice simplement, sans moyens sur ou sous-proportionnés.

D'autres critères seront pris en compte mais ceux-ci sont les principaux. Le total des points s'exprimera sous la forme d'une note sur 100 points, avec 20 points par exercice. Le détail des notes ne sera pas donné à la fin du tournoi.

Pour les exercices où c'est possible, un **set de données de test** vous est fourni. Cela vous permet de vérifier que votre programme fonctionne bien comme il est censé le faire. Attention : lors de l'évaluation de votre programme, d'autres données que celles testées seront utilisées et, par conséquent, votre programme doit s'adapter à toutes les données qui pourront lui être fournies.

### Règles du tournoi :

- Le code doit être rendu sur un lien <https://replit.com> sur le Discord de l'Asynconf (<https://discord.gg/asynconf>) dans le salon #👷 | déposer-son-projet. Toutes les informations dont vous avez besoin pour le rendu du projet sont disponibles en tant que message épinglé sur le salon.
- Le langage utilisé pour réaliser les différents exercices est libre
- Le tournoi est individuel et ne doit donc pas être réalisé à plusieurs.
- L'utilisation d'internet afin de chercher des informations et/ou de l'aide est autorisée.
- Tout code identique trouvé sur deux projets vaudra une disqualification à la fois du copieur et du copié. Veillez à bien garder votre code secret durant la durée des épreuves, afin de ne pas vous faire disqualifier.
- Vous n'êtes pas obligés de faire tous les exercices. Si un exercice vous semble trop dur, ou que vous n'y arrivez pas, n'hésitez pas à passer à l'exercice suivant.
- L'usage de café afin de rester éveillé est autorisée, et recommandée ☕



## Exercice 1 : Le code ensorcelé

Difficulté : Facile

Temps estimé : 15 min.

**Contexte :** Une sorcière de renom cachait souvent son grimoire dans une grande armoire scellée par un sortilège qui lui empêchait de l'ouvrir. Seulement ce sortilège pouvait ouvrir l'armoire. Elle décida donc de l'écrire sur un morceau de parchemin et d'utiliser un autre sort de confusion afin de cacher la formule à la vue de tous. Malheureusement, elle n'arrive plus à se souvenir de son sort, et ne comprend plus les techniques qu'elle a utilisées pour camoufler son parchemin. Vous devrez retrouver la formule lui permettant d'ouvrir son armoire à partir du parchemin encodé.

**Énoncé :** Une amie de la sorcière lui a rappelé les sortilèges qu'elle a utilisés pour cacher sa formule. Elle a décalé toutes les lettres de 2 positions dans l'alphabet, transformant un a en c, sans toucher aux symboles, puis elle a encodé le résultat en base64.

**Entrée :** Vous devrez demander la [formule encodée] de la sorcière.

**Sortie :** Votre programme devra renvoyer la {formule décodée} après avoir effectué les transformations nécessaires afin qu'elle devienne lisible.

**Set de données de test :**

Entrée	Sortie
formule encodée	formule décodée
Q2R0Y2VjZmNkdGMsIG5nIHVxdHZrbsOoaWcgaHd2LCBndiByZ3Rmd3RjICE=	Abracadabra, le sortilège fut, et perdura !
U3djcyGygbuKAmWNOb3FrdGcgdWcgdHF3eHRnLCDDp2MgZmNwdWcgY3cgTnF3eHRnLg==	Quand l'armoire se rouvre, ça danse au Louvre.
TmMgcHrdiBncCDDqXbDqSwgbmMgbndwZyBma3VyY3Rjw652IG5xdHVzd2cgbmdlIG5xd3JlIHVnlG9ndnZncHYgw6AgandObmd0Lg==	La nuit en été, la lune disparaît lorsque les loups se mettent à hurler.
Q3cgTnF3eHRnLCBuYyBud29rw6h0ZyB14oCZY25ud29nIGd2IG7igJljdG9xa3RnIHxigJlxd3h0Zy4=	Au Louvre, la lumière s'allume et l'armoire s'ouvre.



## **Exercice 2 : Les déguisements délirants**

Difficulté : Facile

Temps estimé : 50 min.

**Contexte :** Le soir d'Halloween, un groupe d'enfants va chercher des bonbons dans leur rue déguisés en momies, en vampires et en fantômes. On cherche à savoir le nombre de bonbons récoltés en fonction du costume des enfants.

**Énoncé :** Lorsqu'ils passent par un appartement, jusqu'à 2 enfants en costume de momie récupèrent 3 bonbons chacun, 1 enfant en costume de fantôme en récupère 4 et le reste des enfants ne récupère pas de bonbons.

Lorsqu'ils passent par une maison, tous les enfants avec un costume de fantômes sauf un reçoivent 3 bonbons. Les enfants en costume de momie en reçoivent 1 chacun et jusqu'à 4 vampires reçoivent 2 bonbons, les autres vampires récoltent quant à eux 1 bonbon.

Lorsqu'ils passent par une villa, 1 enfant en vampire reçoit 12 bonbons et le reste des enfants reçoit 2 bonbons chacun.

Quand ils rentrent chez eux, les parents des vampires mangent 2 bonbons par habitation visitée, les parents des fantômes mangent 2 bonbons pour chaque appartement visité et les parents des momies mangent 1 bonbon pour chaque maison visitée.

**Entrée :** Les enfants visiteront [nombre de villas], [nombre de maisons] et [nombre d'apparts.]. Ils seront [nombre de momies], [nombre de vampires] et [nombre de fantômes].

**Sortie :** Votre programme devra renvoyer {bonbons des momies}, {bonbons des vampires} et {bonbons des fantômes}.

**Set de données de test :**

Entrées						Sorties		
nombre de villas	nombre de maisons	nombre d'apparts.	nombre de momies	nombre de vampires	nombre de fantômes	bonbons des momies	bonbons des fantômes	bonbons des vampires
1	1	1	1	1	1	5	4	8
5	7	18	4	8	13	169	418	154
5	12	28	3	8	3	222	158	184
5	14	27	5	8	1	268	64	206



## Exercice 3 : La tournée du quartier

Difficulté : Moyen

Temps estimé : 20 min.

**Contexte :** Un enfant décide de récolter des bonbons dans son quartier, mais comme il est malin il se dit qu'il peut essayer de passer plusieurs fois aux mêmes habitations afin d'avoir le plus de bonbons possible. Cependant, les habitants se rendent compte de sa supercherie. Combien de bonbons aura-t-il à la fin et combien de fois aura-t-il fait le tour du quartier pour obtenir ces bonbons ?

**Entrées :** Dans le quartier de l'enfant, il y a 3 villas, 12 maisons et 23 appartements. Votre programme devra prendre en entrée le nombre de [bonbons par villa], le nombre de [bonbons par maison], le nombre [bonbons par appart.] ainsi que le nombre de [bonbons en moins] que tous les habitants donneront en moins en voyant l'enfant repasser.

**Sorties :** Votre programme retournera le {nombre total de bonbons récoltés} ainsi que le {nombre de tours effectués} avant de ne plus recevoir de bonbons.

**Exemple** avec [bonbons par villa = 17], [bonbons par maison = 14], [bonbons par appart. = 11] et [bonbons en moins = 4] pour 1 logement :

	Villa	Maisons	Appartements
1 <sup>er</sup> tour	17 bonbons	14 bonbons	11 bonbons
2 <sup>e</sup> tour	13 bonbons	10 bonbons	7 bonbons
3 <sup>e</sup> tour	9 bonbons	6 bonbons	3 bonbons
4 <sup>e</sup> tour	5 bonbons	2 bonbons	0 bonbon
5 <sup>e</sup> tour	1 bonbon	0 bonbon	0 bonbon
6 <sup>e</sup> tour	0 bonbon	0 bonbon	0 bonbon

**Set de données de test :**

Entrées				Sorties	
bonbons par villa	bonbons par maison	bonbons par appart.	bonbons en moins	nombre total de bonbons récoltés	nombre de tours effectués
17	14	11	4	1002	5
5	9	7	12	284	1
23	12	5	2	1143	12
14	17	20	6	1480	4



## **Exercice 4 : Des tenues complètes ?**

Difficulté : Moyen

Temps estimé : 30 min.

Contexte : On a une liste de clients ayant effectué des achats et on souhaite savoir si ces derniers ont acheté des déguisements complets pour le soir d'Halloween.

Énoncé : Lors de cet exercice, un fichier JSON vous est fourni. Vous devrez, à partir d'une liste de clients disponibles sur ce fichier, et de leurs achats, déterminer les costumes complets que chaque client s'est procuré.

Les différents articles que les clients ont pu acheter sont :

- Une boîte de pétards
- Des chaînes
- Des feux d'artifices
- Une citrouille
- Une cape
- Des gâteaux
- Du chocolat
- Un drap blanc
- Un set de maquillage
- Des bonbons
- Des dents de vampire
- Un film d'Halloween
- Un balai
- Des bandelettes blanches
- Du faux sang
- Un masque de fantôme
- Un drap rouge

Les costumes de vampires comportent : des dents de vampire, une cape, du faux sang et un set de maquillage.

Les costumes de fantôme comportent : un masque de fantôme, un drap blanc et des chaînes.

Les costumes de momie comportent : des bandelettes blanches, et un set de maquillage.

Entrée : Lien du fichier JSON : <https://api.asynconf.fr/exercice4.json>

Sorties : Votre programme devra renvoyer la {liste des clients et des costumes complets qu'ils ont achetés}.



## **Exercice 5 : Le Cimetière Inconnu**

Difficulté : Difficile

Temps estimé : 1h15min.

Contexte : Le gardien d'un cimetière, a été averti que le cimetière dans lequel il travaille va bientôt être déplacé. Seulement, ce cimetière a une particularité : il n'accueille que des personnages mystiques. Malheureusement plus aucun nom n'est visible sur les pierres tombales, impossible de savoir qui se trouve dans ces dernières. À partir de l'environnement qui entoure les tombes, il est possible de retrouver leur contenu et donc de faciliter le travail du gardien.

Énoncé : Afin de retrouver le contenu des tombes vous devrez récupérer auprès de l'utilisateur les éléments du décor entourant la tombe ainsi que leurs coordonnées, à savoir :

- une rose (Ros)
- une toile d'araignées (TodA)
- un grimoire (Grim)
- une flaque d'eau (FdE)
- de la boue (Bou)
- une citrouille (Citr)
- une bougie (Boug)
- un buisson (Buis)
- un chandelier (Chand)

et ainsi déterminer qui se trouve à l'intérieur de la tombe, à savoir :

- un zombie (Zom)
- un squelette (Squ)
- un vampire (Vamp)
- une momie (Mom)
- un fantôme (Fan)
- une sorcière (Sor)

Conditions :

- Une tombe contient un zombie si elle a de la boue en dessous et des flaques d'eau sur la droite sur la gauche.
- Une tombe contient un squelette si elle a une citrouille au-dessus et une toile d'araignées à gauche et à droite.
- Une tombe contient un vampire si elle a une rose en dessous, un chandelier à droite et à gauche.
- Une tombe contient une momie si elle a un buisson à droite et à gauche et de la boue en dessous.
- Une tombe contient un fantôme si elle a une bougie à droite et à gauche.
- Une tombe contient une sorcière si elle a un grimoire en dessous.

Entrées : Votre programme prendra en entrée une {liste des éléments}. Chaque élément de la liste se présentera sous la forme : {position y}:{position x}-{type d'élément} et tous les éléments seront séparés par des ;



# Asynconf 2021 - Tournoi d'algorithmique

Exemple de grille avec l'entrée associée :

	1	2	3
1			
2	FdE	Tom	FdE
3		Bou	

2:1-FdE;2:2-Tom;2:3-FdE;3:2-Bou

Sorties : Votre programme retournera la {liste des tombes} avec le type de mort-vivant qui y réside sous le format : T-{position y}:{position x}-{type de mort vivant} et les différentes tombes devront être séparées par des ;

Exemple de grille avec la sortie associée :

	1	2	3
1			
2	FdE	Tom	FdE
3		Bou	

T-2:2-Zom

Set de données de test :

Entrées	Sorties
liste des éléments	liste des tombes
1:1-Chand;1:2-Tom;1:3-Chand;1:5-Citr;2:2-Ros;2:3-Tom;2:4-TodA;2:5-Tom;2:6-TodA;3:3-Grim;4:1-Boug;4:2-Tom;4:3-Boug;4:4-Tom;4:5-Boug;	T-1:2-Vamp;T-2:3-Sor;T-2:5-Squ;T-4:2-Fan;T-4:4-Fan;
1:1-Boug;1:2-Tom;1:3-Boug;1:4-Chand;1:5-Tom;1:6-Chand;2:1-FdE;2:2-Tom;2:3-FdE;2:5-Ros;3:1-Tom;3:2-Bou;3:3-Buis;3:4-Tom;3:5-Buis;4:1-Grim;4:4-Bou;	T-1:2-Fan;T-1:5-Vamp;T-2:2-Zom;T-3:1-Sor;T-3:4-Mom;
1:1-Boug;1:2-Tom;1:3-Boug;1:4-Tom;2:1-Buis;2:2-Tom;2:3-Buis;2:4-Grim;3:2-Bou;3:3-Citr;4:2-TodA;4:3-Tom;4:4-TodA;5:1-Chand;5:2-Tom;5:3-Chand;6:2-Ros;	T-1:2-Fan;T-1:4-Sor;T-2:2-Mom;T-4:3-Squ;T-5:2-Vamp;
1:1-Buis;1:2-Tom;1:3-Buis;1:4-Tom;2:1-Tom;2:2-Bou;2:3-Citr;2:4-Grim;3:1-Grim;3:2-TodA;3:3-Tom;3:4-TodA;4:1-Boug;4:2-Tom;4:3-Boug;	T-1:2-Mom;T-1:4-Sor;T-2:1-Sor;T-3:3-Squ;T-4:2-Fan;