

Le jeu des allumettes – Feuille à rendre

Nom et prénom :

Compléter la feuille au fur et à mesure de votre avancement sur le site.

Tout au long de cette feuille, `nb_allumettes` correspond au nombre d'allumettes qui restent, `joueur` représente le numéro du joueur (1 ou 2) dont c'est le tour et `choix` est le nombre d'allumettes qu'il veut prendre.

? EXERCICE 1 :

Le but de ce tableau est de savoir si la valeur demandée par le joueur est une réponse acceptable ou s'il faut lui redemander une nouvelle valeur. Lorsqu'il y a plusieurs valeurs possibles, vous en choisirez une qui est convenable.

nb_allumettes	choix	choix<1	choix>3	choix>nb_allumettes	Redemander?
6	0	True	False	False	oui
6	2	False	False	False	non
6	3				
6	4				
2		False	False	True	oui
5		False	True	True	

- 1) Compléter le tableau ci-dessus.
- 2) Si on regarde les tests `choix<1`, `choix>3` et `choix>nb_allumettes`, dans quel cas est-ce qu'il faut redemander au joueur? Entourer la bonne réponse.
 - a) au moins un est faux
 - b) au moins un est vrai
 - c) ils sont tous faux
 - d) ils sont tous vrais
- 3) Indiquer les opérateurs logiques (**and**, **or**, **not**) qu'il faut rajouter sur la ligne suivante pour qu'on reste dans la boucle tant que le joueur n'a pas donné une réponse valide.

```
while choix < 1 ... choix > 3 ... choix > nb_allumettes:
```

? EXERCICE 2 :

Ces questions ont pour but de vous aider à compléter la fonction `manche`.

- 1) À l'aide des variables de cette fonction, donner une expression permettant de calculer le nombre d'allumettes qu'il restera après le tour du joueur :
.....
- 2) On souhaite déterminer la condition qui permet de savoir s'il faut continuer la partie.
 - a) Quelle est la valeur de `nb_allumettes` lorsque la partie est finie?
 - b) En déduire une expression booléenne qui vaut **True** lorsque la partie n'est pas finie et **False** lorsqu'elle est terminée.
Par exemple, `choix < 1` est une expression booléenne.

? EXERCICE 3 :

On considère le code de la fonction `partie`. On note `score1` le score du joueur 1 et `score2` celui du joueur 2. Donner une expression booléenne permettant de savoir à la fin de la partie si le joueur 1 a gagné:

? EXERCICE 4 :

Le but de cet exercice est de trouver comment mettre en place la stratégie optimale pour l'ordinateur.

- 1) Remplir le tableau ci-dessous avec G pour une position gagnante et P pour une position perdante.

nb_allumettes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Position	P	G	G	G												

- 2) a) S'il reste 9 allumettes, combien faut-il en prendre pour laisser l'autre joueur dans une position perdante?
b) S'il reste 15 allumettes, combien faut-il en prendre pour laisser l'autre joueur dans une position perdante?
3) Le tableau ci-dessous correspond aux restes de la division euclidienne de nb_allumettes par 4.

nb_allumettes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
nb_allumettes%4	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3

- a) Que vaut nb_allumettes % 4, lorsqu'on est en position perdante?
b) En position gagnante, combien faut-il prendre d'allumettes?
Répondre à l'aide d'une formule utilisant nb_allumettes.
.....

? EXERCICE 6 :

On considère le code de la fonction manche_ordi avant de le compléter.

- 1) Quel est le numéro du joueur humain?
2) Quel est le numéro de l'ordinateur?
3) On suppose que la partie est finie. Donner une expression booléenne qui est vraie à la fin de la boucle uniquement lorsque le joueur humain a gagné:

? EXERCICE 7 :

On considère le code de la fonction partie_ordi. On note score1 le score du joueur 1 et score2 celui du joueur 2. Il y a nb_manches manches à jouer.

- 1) Si score1 = 3 et score2 = 1, combien de manches ont été jouées?
2) Donner une expression utilisant les variables de la fonction permettant de savoir combien de manches ont déjà été jouées au début d'une nouvelle manche:
.....
3) Donner une expression booléenne permettant de savoir s'il faut faire une nouvelle manche ou non:

? EXERCICE 9 :

On note à nouveau score1 et score2 les scores des deux joueurs et nb_manches le nombre maximum de manches à jouées.

- 1) Donner une expression booléenne qui devient vraie dès que le joueur 1 est sûr de gagner:
2) Quelles sont les 3 situations possibles qui mettent fin à la partie:
.....
3) Donner l'expression booléenne à mettre comme condition de la boucle **while**.
.....