

Exercice Algo

Partie I

Exercice I

Lors du cours d'astronomie que le professeur donne aux élèves, il leur explique le principe des éclipses, puis il leur indique la date de la prochaine éclipse totale sur Algoréa. Vous connaissez donc la date de l'éclipse ainsi que la date actuelle et vous vous demandez dans combien de jours aura donc lieu cette éclipse. Notez que ces deux dates sont exprimées en nombre de jours du calendrier algoréen.

Ce que doit faire votre programme :

Vous savez que l'éclipse aura lieu le 12581^e jour et que la date actuelle est le 11937^e jour. Votre programme doit calculer et afficher le nombre de jours qu'il faudra patienter avant de pouvoir admirer l'éclipse.

Exercice II

Les élèves ne semblent pas à l'aise avec votre robot : ce n'est pas un être humain, ce n'est pas un animal non plus... ils sont donc très intrigués, voire même parfois un peu inquiets. Afin de les rassurer, vous souhaitez que votre robot distribue 3 bonbons à chaque élève. Pour cela, vous avez besoin de calculer le nombre de bonbons à acheter.

Ce que doit faire votre programme :

L'école est formée de 4 classes, constituées respectivement de 25, 30, 27 et 22 élèves. Cependant, 8 élèves sont absents aujourd'hui. Sachant que chaque élève présent doit recevoir 3 bonbons, écrivez un programme qui calcule puis affiche le nombre total de bonbons nécessaires.

Vous n'avez pas besoin de calculatrice : **effectuez les calculs dans votre programme.**

Exercices III

À la fin d'une course d'endurance, les élèves d'une classe sont épuisés. Leur enseignant leur parle de l'algoréathlon, un exercice d'endurance connu sur toute la planète. Il s'agit d'enchaîner 2 km de natation, 34 km de cyclisme et 6 km de course à pied en un jour, puis de le refaire un deuxième jour, et encore un troisième jour.

Les élèves se demandent quelle distance a été parcourue après chaque jour. Vous profitez de l'occasion pour faire intervenir votre robot.

Ce que doit faire votre programme :

L'algoréathlon se constitue de trois étapes à effectuer chaque jour : 2 km de natation, 34 km de cyclisme et 6 km de course à pied.

Sachant qu'un sportif répète ces trois étapes pendant 3 jours de suite, vous devez afficher la distance totale qu'il a parcourue à la fin du 1^{er} jour, à la fin du 2^e jour, puis à la fin de l'algoréathlon complet.

Afin de rendre l'affichage convivial sur l'écran du robot, vous souhaitez mettre les trois valeurs sur une même ligne, avec une espace entre chaque valeur et la suivante.

Important : pour écrire ce programme, vous devez mémoriser la distance parcourue en un jour en lui donnant un nom, puis utiliser ce nom pour calculer les trois réponses

Exercices IV

La cour de récréation de l'école a bien besoin d'être refaite : d'une part la clôture actuelle est trop rouillée pour résister à un Borlok, et d'autre part il faudrait couvrir le sol en terre par de belles pierres pour éviter d'avoir de la boue dès qu'il pleut.

Le professeur de la classe fait participer ses élèves : il leur demande de calculer l'aire et le périmètre du carré que forme leur cour de récréation. Les élèves n'ayant pas de règle assez longue, ils ont utilisé 4 bâtons de longueurs différentes pour mesurer le côté du carré. Afin d'aider les élèves dans leurs calculs, vous décidez d'écrire un programme.

Ce que doit faire votre programme :

La cour carrée a été mesurée avec quatre bâtons de longueurs respectives 17 m, 7 m, 5 m et 2 m. La longueur du côté de la cour est égale à 5 fois le premier bâton plus 2 fois le second plus 1 fois le troisième plus 2 fois le quatrième.

Votre programme doit afficher deux lignes : la première doit contenir la surface de la cour, et la seconde ligne doit contenir son périmètre. Les résultats doivent être exprimés en mètres carrés et en mètres, respectivement, mais vous ne devez pas afficher l'unité après la valeur numérique.

Important : dans votre programme, commencez par calculer la longueur du côté de la cour et l'enregistrer dans une variable.

Partie II

Exercice I

Créer l'algorithme permettant de donner l'état de l'eau selon sa température. Si l'eau est inférieure ou égale à 0°C alors l'état de l'eau est solide, si la température de l'eau est comprise entre 0 et 100°C alors l'état de l'eau est liquide et si la température de l'eau est supérieure à 100°C alors l'état de l'eau sera gazeux.

Exemple :

Données en entrées : 100

Données en sortie : « L'eau est à l'état solide »

Exercice II

Les habitants de la Comté paient l'impôt selon les règles suivantes :

- Les hommes de plus de 20 ans paient l'impôt
- Les femmes paient l'impôt si elles ont entre 18 et 35 ans
- Les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe du Hobbit, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

Exemples :

Données en entrées : « Homme », 21

Données en sortie : « Imposable »

Données en entrées : « Femme », 36

Données en sortie : « Non imposable »

Exercice III

Cet algorithme est destiné à prédire l'avenir, et il doit être infallible ! Il lira au clavier l'heure, les minutes et les secondes et il affichera l'heure qu'il sera une seconde plus tard. Par exemple, si l'utilisateur tape 21 puis 32 puis 52, l'algorithme doit répondre :

"Dans une seconde, il sera 21 : 32 : 53".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

Exemples :

Données en entrées : 21, 00, 59

Données en sorties : « Dans une seconde il sera 21 : 01 : 00 »

Données en entrées : 21, 59, 59

Données en sortie : « Dans une seconde il sera 22 : 00 : 00 »

Exercice IV

Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux : tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur :

Un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux ans, se voit attribuer le tarif rouge, si toutefois il n'a jamais été responsable d'accident. Sinon, la compagnie refuse de l'assurer.

Un conducteur de moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux ans a le droit au tarif orange s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident, sinon il est refusé.

Un conducteur de plus de 25 ans titulaire du permis depuis plus de deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà

De plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus d'un an.

Écrire l'algorithme permettant de saisir les données nécessaires (sans contrôle de saisie) et de traiter ce problème.

Partie III

Exercice I

Afin d'amuser un groupe d'enfants, vous décidez de leur construire une fusée. Après leur avoir rappelé qu'il ne faut pas le faire sans un adulte, vous sortez un sachet de thé de votre poche et annoncez que ce sera votre fusée ! Incrédules, les enfants vous voient enlever l'agrafe, l'étiquette, vider le thé, et former un cylindre. Il ne reste plus qu'à allumer le haut du cylindre et faire le décompte avant le décollage de la fusée !

Ce que doit faire votre programme :

Votre programme devra lancer le décompte en partant de 100 puis annoncer le décollage, c'est-à-dire afficher une séquence d'annonces de la forme :

```
100
99
...
2
1
0
Décollage !
```

Exercice II

L'automne ayant été très pluvieux, les Bufo Algo, une espèce locale de crapauds, se sont reproduits en grand nombre, et les habitants ont constaté que leur nombre doublait chaque semaine ! Leurs prédateurs naturels, les couleuvres (un type de serpent), sont complètement dépassés !

Avoir trop de crapauds est très gênant (on ne peut plus dormir) ! Les villageois décident donc d'élever un grand nombre de couleuvres avec lesquelles ils pourront contrôler le nombre de crapauds. Il leur faut tout d'abord estimer le nombre de crapauds qu'il y aura au cours des semaines qui viennent. Ils vous demandent votre aide.

Ce que doit faire votre programme :

Sachant qu'il y a actuellement 1337 crapauds et que leur nombre double chaque semaine, votre programme devra afficher le nombre de crapauds qu'il y aura après la 12^e semaine.

Exercice III

C'est la dernière semaine de cours et l'école organise une grande kermesse. L'un des stands, « La foire aux bonbons », propose un jeu permettant de gagner des bonbons. Le jeu est simple : il faut atteindre le plus grand nombre de fois possible une cible sans jamais la rater. On peut gagner gros, car plus on touche la cible, et plus on gagne de bonbons à chaque fois qu'on touche la cible ! Les élèves aimeraient connaître le nombre de bonbons qu'ils peuvent gagner en fonction du nombre de tirs consécutifs réussis.

Ce que doit faire votre programme :

Toucher la cible au premier tir rapporte un bonbon, toucher la cible au deuxième tir rapporte deux bonbons de plus, la toucher au troisième tir rapporte encore trois bonbons de plus, etc. Écrivez un programme qui affiche sur la première ligne le nombre total de bonbons obtenus si l'on ne réussit qu'1 tir, puis qui affiche sur la deuxième ligne le nombre de bonbons récupérés si l'on réussit 2 tirs de suite, puis sur la troisième ligne le nombre de bonbons récupérés si l'on réussit 3 tirs de suite, etc. jusqu'à la valeur que l'on peut récupérer si l'on réussit 50 tirs de suite.

Exemple :

Si le programme, s'arrêtait à 5 tirs il devra afficher :

```
1
3
6
10
15
```

Exercice IV

Les enfants de la classe de maternelle décident de construire une très grande tour à l'aide de petits cubes en bois. Ils savent exactement la forme qu'ils souhaitent pour leur tour, mais ils n'arrivent pas à savoir s'ils auront suffisamment de cubes pour la construire. Ils vous demandent de les aider à calculer le nombre de cubes nécessaires.

Ce que doit faire votre programme :

L'objectif est de construire une tour à l'aide de petits cubes en bois, sachant que la forme de cette tour consiste en un ensemble de grands cubes placés les uns au-dessus des autres. La base de la tour est un cube de taille $17 \times 17 \times 17$, c'est-à-dire composé de $17 \times 17 \times 17 = 4\,913$ petits cubes. Sur ce cube est posé un autre cube de taille $15 \times 15 \times 15$. Au-dessus de ce dernier se trouve un cube de $13 \times 13 \times 13$. La tour continue ainsi jusqu'à atteindre le sommet, qui consiste en un cube de taille $1 \times 1 \times 1$.

Votre programme doit calculer et afficher le nombre total de petits cubes nécessaires pour construire la pyramide. Effectuez les calculs dans le programme en y intégrant une boucle.

Exercice V

C'est l'heure du cours de mathématiques et aujourd'hui les enfants vont travailler la multiplication. Malheureusement, l'institutrice ne retrouve que la petite table de multiplication, qui va jusqu'à 5 fois 5, mais pas la grande table, qui va jusqu'à 20 fois 20. Elle souhaiterait que vous lui imprimiez une nouvelle table allant jusqu'à 20 fois 20, pour qu'elle puisse l'afficher au mur.

Ce que doit faire votre programme :

Voici à quoi ressemble la table de multiplication allant jusqu'à 5 fois 5.

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

Exercice VI

À l'aide d'une feuille de brouillon, essayez de déterminer le résultat affiché par le programme suivant.

```
const nbPersonnes = 3, nbJours = 4
let prixSejour = 0, prixJournee = 1

for(let i = 0 ; i < nbJours ; i++){
    prixSejour = prixSejour + prixJournee
    prixJournee = prixJournee + 2
}

prixSejour = prixSejour + 14
console.log(`Prix par personne : ${Math.round(prixSejour / nbPersonnes)}`)
```

Exercice VII

Ecrire un algorithme qui calcul et affiche la moyenne de n nombres saisis au clavier.

Exercice VIII

Ecrire un algorithme qui lit la suite de « n » prix en entier des achats d'un client (Les prix sont saisis par l'utilisateur). Calculer le montant à payer, lire le montant versé qu'il paye et simuler la remise de la monnaie.

Attention on ne peut pas simuler la remise de la monnaie si le montant versé est inférieur au montant à payer.

La personne ne peut pas s'enfuir sans payer

Exemples :

Données en entrées : 25(somme des prix saisis par l'utilisateur), 30

Données en sortie : « 1 x Billet de 5€ »

Données en entrées : 32(sommes des prix saisis par l'utilisateur), 50

Données en sortie : « 1 x Billet de 10€, 1 x Billet de 5€, 1 x Pièce de 2€, 1 x Pièce de 1€ »

BONUS

Exercice CXXIII

L'auberge dans laquelle vous avez prévu de passer la nuit ce soir propose des tarifs très intéressants, pour peu que l'on n'arrive pas trop tard. En effet, plus on arrive tôt moins on devra payer. Vous essayez de construire un programme vous donnant directement le prix à payer en fonction de votre heure d'arrivée.

Ce que doit faire votre programme :

Votre programme lira un entier, l'heure d'arrivée, qui sera compris entre 0 et 12 inclus. 0 correspond à midi, 1 à 1h de l'après-midi, etc. et 12 à minuit.

Le prix de la chambre est de 10 pièces à midi, et augmente de 5 pièces chaque heure après midi. Il est donc de 15 pièces à 13h, etc. Il ne peut cependant pas dépasser 53 pièces.

Votre programme devra afficher le prix à payer correspondant à l'heure d'arrivée donnée.

Exemples :

Donnée en entrée : 7

Donnée en sortie : 45

Donnée en entrée : 10

Donnée en sortie : 53

Exercice CXII

À peine arrivé dans le village, voilà qu'une bagarre générale est sur le point d'éclater ! Tout en vous mettant à l'abri, vous tâchez de savoir ce qui se passe. On vous explique que le village est principalement composé de deux grandes familles rivales qui ne se supportent pas. Tout sujet étant une source de discorde possible, ils avaient décidé que les superficies de leurs champs respectifs ne devaient pas être trop différentes afin de ne pas attiser la jalousie de la famille opposée.

Mais voilà que le patriarche des Arignon suspecte qu'un des champs des Evaran est trop grand ! Vous décidez de les aider ; mais la tâche ne sera pas facile, chacun gardant jalousement secrète la superficie réelle de ses champs.

Ce que doit faire votre programme :

Votre programme devra lire deux entiers, la superficie d'un champ des Arignon et la superficie d'un champ des Evaran. Si l'un des champs est plus grand d'au moins 10 m² (strictement) que l'autre champ, alors il faudra afficher le texte « La famille X a un champ trop grand », « X » devant bien sûr être remplacé par « Arignon » ou « Evaran » selon le cas.

Exemples :

Données en entrée : 42, 54

Données en sortie : « La famille Evaran a un champs trop grand »

Données en entrée : 10, 20

Données en sortie :

Exercice MCCXVI :

Alors que vous traversez une forêt vous ne pouvez vous empêcher d'admirer la végétation autour de vous et notamment les nombreuses espèces d'arbres. Malgré votre intérêt, vous êtes très mauvais botaniste et avez beaucoup de mal à identifier les différents arbres. Une personne que vous croisez vous donne quelques indications et vous décidez d'écrire un programme qui vous donnera le nom de l'arbre en fonction de ses caractéristiques.

Ce que doit faire votre programme :

Il existe 4 types d'arbres :

- le "Tinuviel" fait moins de 5 mètres de haut et ses feuilles sont composées de plus de 8 folioles
- le "Calaelen" fait plus de 10 mètres de haut et ses feuilles sont composées de plus 10 folioles
- le "Falarion" fait moins de 8 mètres de haut et ses feuilles sont composées de moins de 5 folioles
- le "Dorthonion" fait plus de 12 mètres de haut et ses feuilles sont composées de moins de 7 folioles

Votre programme lira deux entiers, la hauteur et le nombre de folioles de l'arbre, et affichera le nom de l'arbre correspondant.

Toutes les inégalités sont à prendre au sens large, c'est-à-dire que "moins" signifie "moins ou égal" ou et "plus" signifie "plus ou égal".

Exercice XXII :

L'auberge dans laquelle vous vous arrêtez pour la nuit adapte ses prix en fonction de l'âge du client et du poids de ses bagages. Les règles ne vous étant pas très claires, vous décidez d'écrire un petit programme qui vous permettra facilement, à vous et à vos compagnons de voyage, de connaître le prix d'une nuit.

Ce que doit faire votre programme :

Une chambre ne coûte rien si on a 60 ans (l'âge de l'aubergiste !) et 5 écus si on a strictement moins de 10 ans. Pour les autres personnes c'est 30 écus plus un supplément de 10 écus si on a au moins 20 kilos de bagages.

Votre programme doit lire deux entiers, l'âge et le poids des bagages de la personne et doit afficher le prix, sous la forme d'un entier.

Exercice XXXII :

Le dimanche, de nombreux enfants sont présents au sein de la bibliothèque avec leurs parents. La grande mode ces derniers temps chez les enfants est de calculer leur « nombre d'amour » pour savoir s'ils sont amoureux (ou pas) d'un autre enfant. Si deux enfants ont le même nombre alors ils sont amoureux.

Intrigué, vous décidez d'écrire un programme permettant de calculer rapidement le « nombre d'amour » correspondant à un prénom donné.

Pour calculer ce nombre, on commence par convertir chaque lettre du prénom en un nombre (A vaut 0, B vaut 1...) puis on calcule la somme de ces nombres. Si le résultat est strictement plus petit que 10, alors on a trouvé le « nombre d'amour ». Sinon, on remplace le nombre par la somme de ses chiffres. On continue ainsi tant que le résultat est plus grand ou égal à 10.

La fonction `String.fromCharCode(number)` retourne le caractère associé au nombre de la table ASCII

Ex : `String.fromCharCode(97) // « a »`

La fonction `String.charCodeAt(index)` retourne le nombre associé au caractère de table ASCII

Ex : `« a ».charCodeAt() // 97`

Ex :

```
let phrase = « The quick brown fox jumps over the lazy dog. »
```

```
phrase.charCodeAt(4) // 113(q)
```

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/String/charCodeAt
https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/String/fromCharCode
https://fr.wikibooks.org/wiki/Les_ASCII_de_0_%C3%A0_127/La_table_ASCII

Exemples :

Donnée en entrée : ADA GWOAG

Donnée en sortie : 3 3

Exercice MCCXV :

Votre programme devra lire un tableau d'entiers. La première entrée du tableau est le nombre total de positions sur la table suivie du nombre de changements de positions. Ensuite on retrouvera pour n positions un entier représentant le numéro de la personne, qui doit s'installer à cette position.

Puis pour n changement de position Il faut lire ensuite les changements exprimés sous la forme de deux entiers chacun : position1 et position2. Un changement (position1, position2) signifie que les deux personnes qui étaient à ses positions doivent échanger leurs places (les positions sont indexées à partir de 0).

Vous devrez retourner un tableau d'entier, avec pour chaque position, le numéro de la personne qui s'y trouve une fois tous les changements faits.

Exemples :

Donnée en entrée : [5,3,1,2,3,4,5,1,2,1,3,4,0]

Donnée en sortie : [5,4,2,3,1]

Evolution des numéros dans l'exemple :

Au début : 1,2,3,4,5

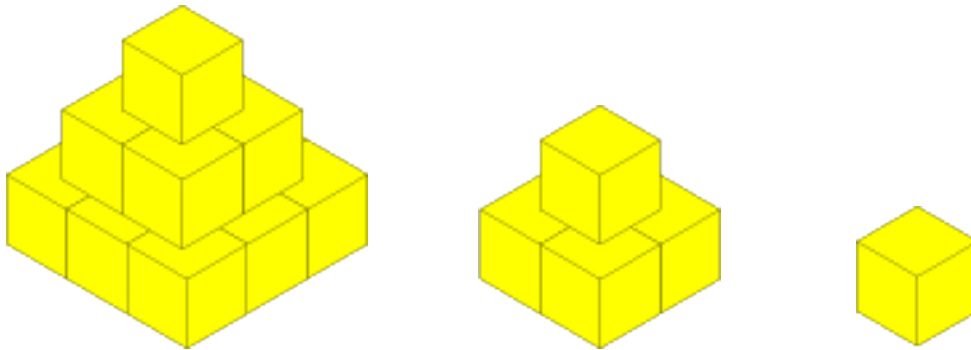
Après le changement (1,2) : 1,3,2,4,5

Après le changement (1,3) : 1,4,2,3,5

Après le changement (4,0) : 5,4,2,3,1

Exercice 1984 :

Les habitants adorent les constructions en forme de pyramide ; de nombreux bâtiments officiels ont d'ailleurs cette forme. Pour fêter les 150 ans de la construction de la ville, le gouverneur a demandé la construction d'une grande et majestueuse pyramide à l'entrée de la ville. Malheureusement, en ces périodes de rigueur budgétaire, il y a peu d'argent pour ce projet. Les architectes souhaitent cependant construire la plus grande pyramide possible étant donné le budget prévu.



Ce que doit faire votre programme :

Votre programme doit d'abord lire un entier : le nombre maximum de pierres dont pourra être composée la pyramide. Il devra ensuite calculer et afficher un entier : la hauteur de la plus grande pyramide qui pourra être construite, ainsi que le nombre de pierres qui sera nécessaire.

Exemples :

Donnée en entrée : 20
Donnée en sortie : 3 14

Donnée en entrée : 26042
Donnée en sortie : 42 25585

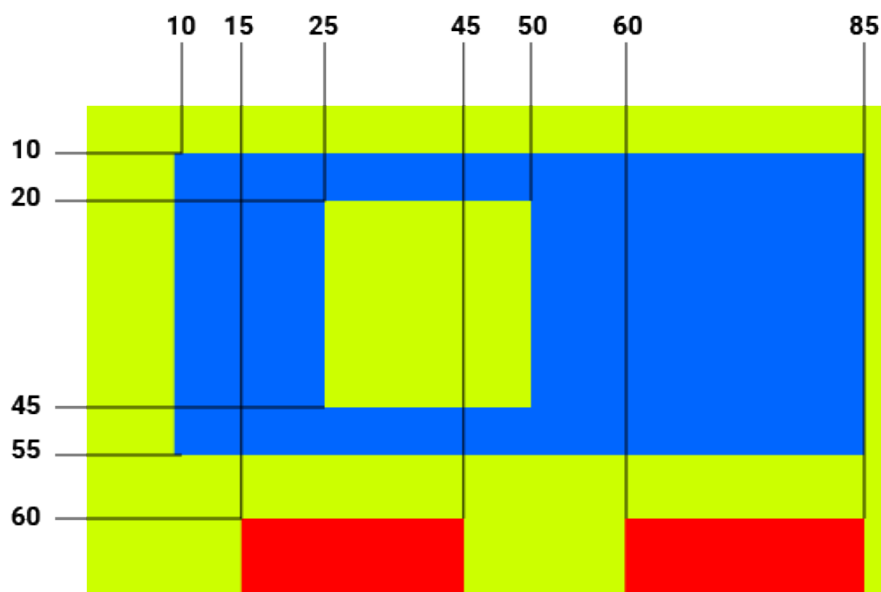
Exercice 007 :

Un espion a été démasqué dans la ville où vous vous trouvez. Son interrogatoire n'a pas été très fructueux : la seule chose que vous savez, c'est qu'il espionnait les savants de l'université, une puissance étrangère étant intéressée par leurs recherches. Vous vous rendez donc à l'université pour discuter avec les chercheurs mais à peine arrivé, vous êtes recruté comme assistant par le laboratoire d'étude du comportement humain.

Celui-ci réalise une expérience consistant à demander à plusieurs personnes de placer chacune un jeton sur une table contenant des zones de différentes couleurs. Les chercheurs souhaitent ainsi étudier si le choix de la zone où une personne place son jeton est lié à la couleur des vêtements qu'elle porte.

Ce que doit faire votre programme :

Sur une table est placée une feuille de papier rectangulaire de 90 cm de large et 70 cm de haut, composée de zones de différentes couleurs, comme le décrit la figure ci-dessous. Un certain nombre de personnes placent l'une après l'autre un jeton où elles le souhaitent sur la table, à l'exception des frontières entre les différentes zones.



On vous donne en entrée un tableau d'entier qui représente les coordonnées des jetons. La première entrée du tableau est le nombre de jetons, ensuite pour n nombre de jetons on retrouve les coordonnées x et y.

Votre programme devra qualifier chaque jeton avec l'un des textes suivants, en fonction de la couleur sur laquelle il se trouve :

- « En dehors de la feuille »
- « Dans une zone jaune »
- « Dans une zone bleue »
- « Dans une zone rouge »

Essayez d'écrire votre programme de sorte qu'il y ait au maximum une condition par possibilité de texte affiché.

Exemples :

Donnée en entrée : [4,16,12,30,22,64,62,-5,86]

Donnée en sortie : ['Dans une zone bleue', 'Dans une zone jaune', 'Dans une zone rouge', 'En dehors de la feuille']

Exercice 451 :

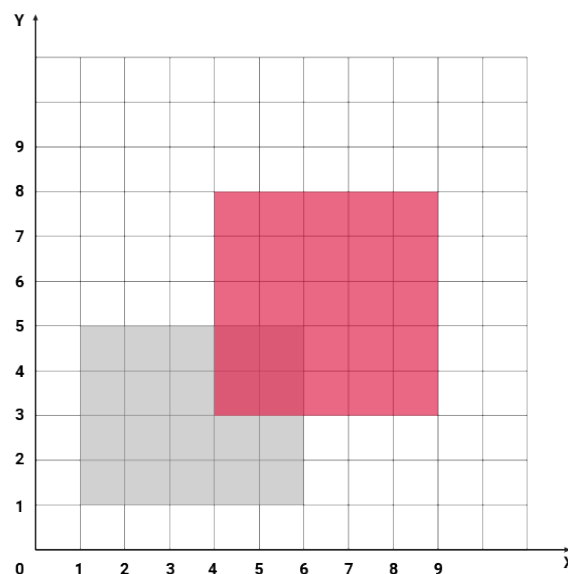
La ville comprend de nombreuses casernes de pompiers et chacune a sa propre zone d'intervention qui lui est réservée. Cependant en regardant ces zones il vous semble qu'elles ne sont pas très bien choisies car parfois elles se recoupent alors que certains endroits de la ville sont en dehors de toutes les zones et donc ne sont pas protégées par les pompiers. Vous décidez d'aider le maire à mieux organiser l'action des pompiers.

Ce que doit faire votre programme :

Votre programme doit lire la description de plusieurs paires de zones rectangulaires, et pour chacune, déterminer si les deux rectangles s'intersectent.

Vous devez lire un tableau d'entiers, le premier entier correspond au nombre de paires de zones que votre programme devra tester. Ensuite, pour chaque paire possible, deux zones rectangulaires et parallèles aux axes vous sont données l'une après l'autre. Chaque zone est décrite par 4 entiers : son abscisse minimale et maximale puis son ordonnée minimale et maximale.

Sur cet exemple, la zone du bas est donc décrite par les 4 entiers (1, 6, 1, 5) et l'autre par (4, 9, 3, 8) :



Pour chaque paire de zones, votre programme doit définir si les zones s'intersectent. Vous devez retourner un tableau de booléen pour chaque paire tester. True si elles s'intersectent False sinon.

Exemples :

```
Donnée en entrée : [  
  [  
    [1,6,1,5],[4,9,3,8]  
  ]  
]  
Donnée en sortie : [true]
```

Exercice CAESAR :

Alors que vous avez enfin trouvé le livre que vous cherchiez, il s'avère qu'il était complètement chiffré. Vous avez déjà réussi à déchiffrer la première page du livre, et celle-ci vous a indiqué le système de chiffrement utilisé dans le reste du livre.

La clé de chiffrement est ici un simple nombre, qu'on appelle D, pour « décalage ». On remplace alors chaque lettre de l'alphabet par la lettre située D places plus loin dans l'alphabet, considéré de manière circulaire.

Ainsi, si le décalage est de 2, alors :

- A devient C
- B devient D
- ...
- X devient Z
- Y devient A
- Z devient B

La clé utilisée pour chiffrer le texte change à chaque page et pour la page numéro X elle vaut

- $3 * X$ si X est pair
- $(-5) * X$ si X est impair

À vous de déchiffrer tout le livre !

Vous devez lire un tableau, la première entrée correspond au nombre de page. Pour les nbPages – 1 entrée suivantes contiennent chacune le texte des pages numéro 2,3,4... nbPages.

Le texte peut contenir des lettres, chiffres, ou caractères de ponctuation, mais pas d'accents.

Vous devez afficher le texte déchiffré pour chacune des pages.

Chaque lettre codée doit être remplacé par la lettre décodée. Les autres caractères (ponctuation, '_', espaces, chiffres), sont laissés tels quels.

Vous devez respecter la casse : si une lettre était en majuscule (ou minuscule), elle doit le rester !

Exemples :

Donnée en entrée : [

```
4,  
«lkio kyz rg ykiutjk vgmk ja robxk » ,  
«Npeep alrp pde wl alrp yfxpcz 3 » ,  
« Qf hauou pazo xm cgmfdudyq bmsq ! »
```

]

Donnée en sortie : [

```
« Ceci est la seconde page du livre »,  
« Cette page est la page numero 3 »,  
« Et voici donc la quatrieme page ! »
```

]

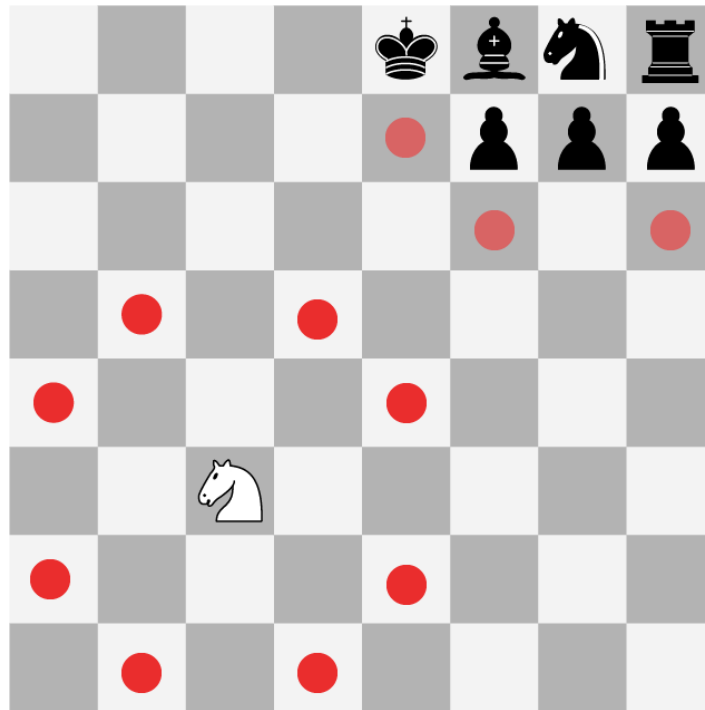
Pour tester si un caractère est une lettre, vous pouvez utiliser cette fonction, présente dans le fichier « index.js » pour les exercices bonus :

```
function isAlpha(caractere) {  
    return /^[A-Z]$/i.test(caractere)  
}
```

Exercice Chess :

Vous devez écrire un programme qui détermine dans une partie d'échecs, si un cavalier peut prendre une pièce de l'adversaire.

Rappelons que les échecs se jouent sur un plateau carré de 8 cases par 8. Un cavalier se déplace en avançant d'une case horizontalement ou verticalement, puis en allant en diagonale, sans revenir en arrière. Le cavalier peut se déplacer même si la case intermédiaire de son déplacement contient une pièce (amie ou ennemie). La figure suivante montre toutes les possibilités de déplacements de deux cavaliers :



On vous donne en entrée la description du plateau sous forme de tableau de 8x8. Les lettres majuscules représentent les pièces blanches, et les minuscules les pièces noires, les « . » représentent les cases vides. Les cavaliers sont représentés par la lettre « c » (ou « C »), les autres pièces par d'autres lettres de l'alphabet

Votre programme doit retourner « true » si l'un des cavaliers blancs peut se déplacer vers une case contenant une pièce adverse (noire, donc la prendre) et « false » dans le cas contraire.

Exemple d'entrée ci-dessous :


```
const chessboard = [
  ['t', 'c', '.', 'd', 'r', 'f', '.', 't'],
  ['p', 'p', 'p', '.', 'p', 'p', 'p', 'p'],
  ['.', '.', '.', 'p', '.', '.', '.', 'c'],
  ['.', '.', '.', '.', '.', 'f', '.', '.'],
  ['.', '.', 'C', '.', '.', 'P', '.', '.'],
  ['.', '.', 'P', '.', 'D', '.', 'P', '.'],
  ['P', 'P', '.', '.', '.', '.', '.', 'P'],
  ['T', '.', 'F', '.', 'R', 'F', 'C', 'T']
]
```

En sortie nous devons avoir « true »