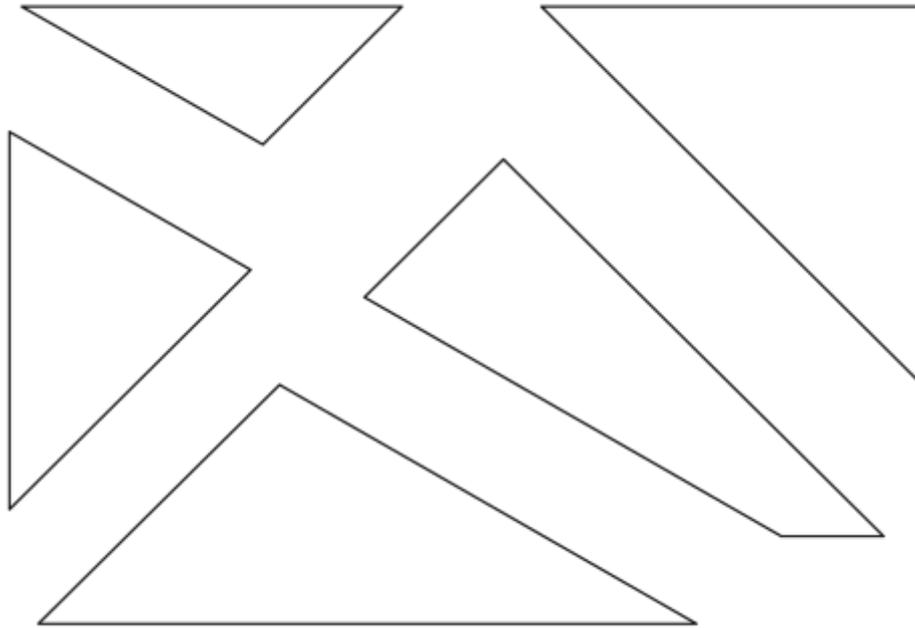


DÉFI N°145 – 1 « LE PUZZLE DE GRENOBLE »



Avec les cinq pièces de ce puzzle, réalise

- un carré,
- un parallélogramme,
- un losange,
- un triangle,
- un trapèze.

Quelle est la nature du polygone à partir duquel ce puzzle a été construit ?

DÉFI N°145 – 2 « LE JOUR PYTHAGORICIEN »

Avez-vous remarqué que le 16 décembre 2020 (c'est-à-dire 16/12/20) était un jour pythagoricien ? En effet, $16^2 + 12^2 = 20^2$.

Quand sera le prochain jour pythagoricien ? Combien y en a-t-il dans notre siècle ? et lesquels ?

Rappel

Un triplet de nombres entiers positifs $(u ; v ; w)$ est pythagoricien si et seulement si $u^2 + v^2 = w^2$.

DÉFI ALGORITHMIQUE N°145

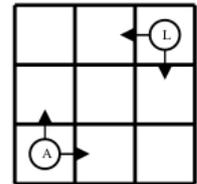
Certaines énigmes du rallye mathématique de Lorraine auraient certainement été plus simples à résoudre à l'aide d'un petit programme informatique.

Nous vous proposons ici, comme défi, de résoudre l'exercice ci-dessous à l'aide d'un programme. L'énoncé était la question subsidiaire de 2009.

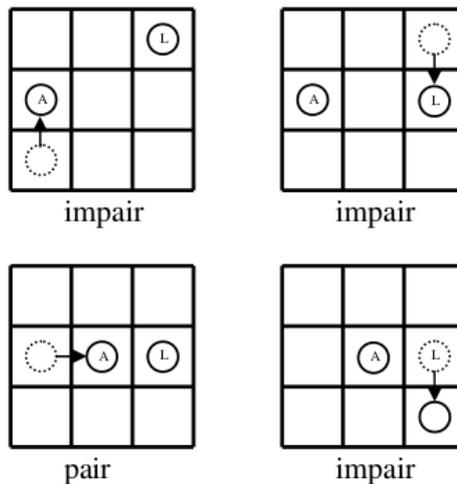
Le loup et l'agneau

Un agneau (A) et un loup (L) sont placés à deux angles opposés d'un quadrillage. En commençant par l'agneau, ils avancent à tour de rôle d'une case en suivant les flèches (vers la droite et vers le haut pour l'agneau, vers la gauche et vers le bas pour le loup). Pour décider de la direction à prendre, on joue avec un dé :

- l'agneau avance d'une case vers la droite si le dé indique un nombre pair, et d'une case vers le haut si le dé indique un nombre impair ;
 - le loup avance d'une case vers la gauche si le dé indique un nombre pair, et d'une case vers le bas si le dé indique un nombre impair.
- Si le loup et l'agneau arrivent dans la même case, le loup capture l'agneau.



Voici le déroulement d'une partie (où l'agneau échappe au loup) :



La règle du jeu est la suivante :

- le loup gagne deux euros s'il capture l'agneau ;
- l'agneau gagne un euro s'il échappe au loup.

Le jeu favorise-t-il l'un des joueurs ou pas ?

L'objectif est de proposer une fonction qui crée des échantillons de plusieurs parties pour déterminer les proportions des parties où le loup gagne et des parties où l'agneau gagne. Cette fonction suppose de simuler des parties.

Le problème est relativement conséquent et suppose un découpage en sous-problèmes.

ÉLÉMENTS DE SOLUTION DU DÉFI N°144 – 1

« UN CARRÉ MAGIQUE POUR 2021 »

Le défi

3	14		
16		11	
	4	18	7
10	15		12

Complète ce carré pour que les sommes des quatre nombres de chaque ligne, de chaque colonne et chaque diagonale soient égales. Quels sont les carrés magiques où on peut placer le nombre **2021** ?

Éléments de solution

3	14	t	v
16	z	11	u
y	4	18	7
10	15	x	12

Les expressions algébriques qui suivent sont égales.

$$37 + x$$

$$29 + y$$

$$27 + z + u$$

$$17 + t + v$$

$$33 + z$$

$$29 + t + x$$

$$19 + u + v$$

$$25 + v$$

La première et la sixième expression me permettent d'affirmer $t = 8$.

La troisième et la cinquième expression me permettent d'affirmer $u = 6$.

3	14	8	v
16	z	11	6
y	4	18	7
10	15	x	12

Les quatre expressions de chaque colonne sont égales.

$$25 + v$$

$$33 + z$$

$$37 + x$$

$$29 + y$$

$$25 + (x + 12)$$

$$33 + (x + 4)$$

$$37 + x$$

$$29 + (x + 8)$$

3	14	8	x+12
16	x+4	11	6
x+8	4	18	7
10	15	x	12

Chaque valeur de x fournit une solution au carré magique. 2021 peut être égal x ou à $x+4$ ou à $x+8$ ou à $x+12$.

Voici les quatre carrés magiques contenant le nombre **2021**.

3	14	8	2033
16	2025	11	6
2029	4	18	7
10	15	2021	12

3	14	8	2025
16	2017	11	6
2021	4	18	7
10	15	2013	12

3	14	8	2029
16	2021	11	6
2025	4	18	7
10	15	2017	12

3	14	8	2021
16	2013	11	6
2017	4	18	7
10	15	2009	12

Ce carré magique à compléter a été initialement repéré dans un cahier de vacances proposé à des enfants en fin de sixième. Ceux-ci n'ont alors à leur disposition que des méthodes par essais erreurs.

Noël Lambert pense que l'énoncé peut être interprété de plusieurs façons et nous propose [une solution à l'aide de GeoGebra](#).

DES SOLUTIONS DU DÉFI N°144 – 2



Dans ce [calendrier de l'Avent](#) utilisé en 2019 sur notre site, la case « 15 » est voisine de la case « 16 », la case « 7 » est voisine des cases « 8 » et « 6 ».

Comment créer un tel calendrier de « l'Avent » tel que les cases de deux jours consécutifs ne soient voisines ni horizontalement, ni verticalement, ni en diagonale ?

1	9	17	4	12	20
5	13	21	8	16	24
7	15	23	6	14	22
2	10	18	3	11	19

Le procédé de remplissage des deux colonnes rouges a été repris pour les deux colonnes vertes et les deux colonnes violettes.

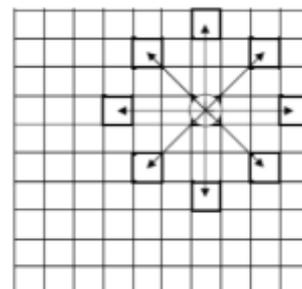
1	3	5	7	9	11
13	15	17	19	21	23
2	4	6	8	10	12
14	16	18	20	22	24

Le procédé de remplissage des deux lignes rouges a été repris pour les deux lignes vertes.

Sont envisageables les déplacements d'Elton ([le kangourou de Raon-l'Étape](#)).

Rappel des déplacements du kangourou

24	1	10	7	4	13
16	19	22	15	18	21
11	8	5	12	9	6
23	2	17	20	3	14



SOLUTION DU DÉFI ALGORITHMIQUE N°144

L'année 2013 est la première année qui peut s'écrire avec 4 chiffres différents depuis 1987. La dernière date que l'on pouvait écrire avec 8 chiffres différents était le 25/06/1987.

Quelle sera la prochaine date que l'on pourra écrire avec 8 chiffres différents ?

Proposez une fonction qui, pour une année n donnée en entrée, renvoie la prochaine date qui s'écrit avec 8 chiffres différents, à partir du premier janvier de l'année n .

Le défi algorithmique du PV 144 demandait de trouver la prochaine date que l'on pourra écrire avec 8 chiffres différents. La fonction **prochaine_date** renvoie la date cherchée. Elle fait appel à la fonction **différents** qui vérifie que les chiffres d'une date sont tous différents et à la fonction **bissextile** qui teste si une année est bissextile.

Effectuer `prochaine_date(26,6,1987)` permet d'obtenir la réponse cherchée.

Pseudo-code

Fonction différents(j,m,a : entiers ; diff : booléen)

si j < 10 :

 jour ← "0" + chaîne(j) ; *l'écriture du jour est mise au format chaîne*

sinon : *avec ajout d'un zéro pour les entiers inférieurs à 10*

 jour ← chaîne(j) ;

finSi ;

si m < 10 :

 mois ← "0" + chaîne(m) ; *l'écriture du mois est mise au format chaîne*

sinon : *avec ajout d'un zéro pour les entiers inférieurs à 10*

 mois ← chaîne(m) ;

finSi ;

 date ← jour+mois+chaîne(a) ; *mise au format chaîne de la date*

 diff ← Vrai ; *par défaut, on suppose tous les chiffres différents*

pour i allant de 1 à 8, **faire** : *on parcourt la chaîne date*

 car ← date[i] ; *chiffre de la date en cours*

si car **est dans** date[i+1:8], **alors** : *on teste si le chiffre en cours n'est pas*

 diff ← Faux ; *dans la fin de la date*

finSi ;

finPour ;

renvoyer Faux

Fonction bissextile(a:entier;booléen)

renvoyer a=0[4] et (a≠0[100] ou a=0[400])

Fonction prochaine_date(jj,mm,aaaa:entiers;jour,mois,annee : entiers)

 jourMois ← [31,28,31,30,31,30,31,31,30,31,30,31] ; *nombre de jours des mois*

 jour ← jj+1 ;

 mois ← mm ;

 annee ← aa ;

Tant que non(différents(jour,mois,annee)), **faire** :

 jour ← jour + 1 ; *on augmente les jours jusqu'à trouver la date cherchée*

si jour > jourMois[mois], **alors** : *si le numéro du jour dépasse*

si non(mois=2 et jour=28 et bissextile(annee)), **alors** :

 jour ← 1 ; *cas du 29 février*

 mois ← mois + 1 ;

finSi ;

si mois>12, **alors** : *si on arrive à la fin de l'année*

 mois ← 1 ;

 annee ← annee + 1 ;

finSi ;

renvoyer jour,mois,annee

On suppose qu'on entre une date inférieure au 05/12/9876.

Python

```
def differents(j,m,a):
    """
    Fonction differents(j,m,a : entiers; diff : booléen)
    renvoie Vrai si tous les chiffres de la date jma sont      différents, Faux sinon
    """
    if j<10:
        jour="0"+str(j)
    else:
        jour=str(j)
    if m<10:
        mois="0"+str(m)
    else:
        mois=str(m)
    date=jour+mois+str(a)
    diff=True
    for i in range(8):
        car=date[i]
        if car in date[i+1:9]:
            diff=False
    return diff

def bissextile(a):
    """
    Fonction bissextile(a : entier;booléen)
    renvoie Vrai si a est une année bissextile, Faux sinon
    """
    return a%4==0 and (a%100!=0 or a%400==0)

def prochaine_date(jj,mm,aaaa):
    """
    Fonction prochaine_date(jj,mm,aaaa:entiers; jour, mois,  annee : entiers)
    renvoie la prochaine date, après jj/mm/aaaa, dont tous les      chiffres sont différents
    """
    joursMois=[31,28,31,30,31,30,31,31,30,31,30,31]
    jour=jj+1
    mois=mm
    annee=aaaa
    while not(differents(jour,mois,annee)):
        jour=jour+1
        if jour>joursMois[mois-1]:
            if not(mois==2 and jour==28 and bissextile(annee)):
                jour=1
                mois=mois+1
            if mois>12:
                mois=1
                annee=annee+1
    return jour,mois,annee
```