

**DÉFI N°144 – 1**

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 ?

**Remarques**

Dans l'exemple précédent la permutation des cases « 7 » et « 15 » montre que de tels calendriers existent.

**Pour des essais**


## DÉFI N°144 – 2

### « UN CARRÉ MAGIQUE POUR 2021 »

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** ?

<b>3</b>	<b>14</b>		
<b>16</b>		<b>11</b>	
	<b>4</b>	<b>18</b>	<b>7</b>
<b>10</b>	<b>15</b>		<b>12</b>

## DÉFI ALGORITHMIQUE N° 144

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 avait été donné en 2013.

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  $n$ .

**SOLUTION DÉFI N°143 – 1 « DEUX ADDITIONS »**

$$\begin{array}{r}
 \cdot \quad \cdot \quad \cdot \\
 + \\
 \hline
 \cdot \quad \cdot \quad \cdot \\
 \cdot \quad \cdot \quad \cdot
 \end{array}$$

En utilisant tous les chiffres 1, 2, 3, 4, 5, 6, 7, 8 et 9 une seule fois, complète l'addition ci-contre.

$$\begin{array}{r}
 2 \quad 3 \quad 4 \\
 + 6 \quad 5 \quad 7 \\
 \hline
 8 \quad 9 \quad 1
 \end{array}$$

**Papier et crayon** nous permettent d'explorer le sujet.

On remarquera que la différence des deux nombres ajoutés vaut **423**.

On trouve d'autres solutions avec un premier nombre formé par les chiffres 2, 3, 4 et un second formé par 5, 6, 7 (la commutativité de l'addition permet de changer la position de chacun des nombres).

$$\begin{array}{r}
 2 \quad 3 \quad 4 \\
 + 6 \quad 5 \quad 7 \\
 \hline
 8 \quad 9 \quad 1
 \end{array}
 \qquad
 \begin{array}{r}
 2 \quad 4 \quad 3 \\
 + 5 \quad 7 \quad 6 \\
 \hline
 8 \quad 1 \quad 9
 \end{array}
 \qquad
 \begin{array}{r}
 2 \quad 4 \quad 3 \\
 + 6 \quad 7 \quad 5 \\
 \hline
 9 \quad 1 \quad 8
 \end{array}$$

Différence des 2 termes	<b>423</b>	<b>333</b>	<b>432</b>
----------------------------	------------	------------	------------

$$\begin{array}{r}
 3 \quad 2 \quad 4 \\
 + 5 \quad 6 \quad 7 \\
 \hline
 8 \quad 9 \quad 1
 \end{array}
 \qquad
 \begin{array}{r}
 3 \quad 2 \quad 4 \\
 + 6 \quad 5 \quad 7 \\
 \hline
 9 \quad 8 \quad 1
 \end{array}
 \qquad
 \begin{array}{r}
 3 \quad 4 \quad 2 \\
 + 5 \quad 7 \quad 6 \\
 \hline
 9 \quad 1 \quad 8
 \end{array}$$

Différence des 2 termes	<b>243</b>	<b>333</b>	<b>234</b>
----------------------------	------------	------------	------------

On remarquera que la différence des deux nombres ajoutés est **333** ou bien un nombre dont les chiffres sont ceux du premier terme soit 2, 3 et 4.

- En prenant deux termes formés de cette même manière, pas d'aboutissement.
- En prenant deux termes aléatoirement formés des chiffres sus cités, pas d'aboutissement. Ce qui ne signifie nullement que ce ne soit pas possible.

Un **programme informatique** nous indique que 336 solutions répondent au premier problème. 96 solutions répondent au second problème. Par exemple  $269+784=1053$ .

Un programme qui généralise le problème est en [fichier joint](#) sur notre site.

Un programme plus modeste se trouve à la page suivante.

**Programme Python**

```

N=list(range(0,10)) #Liste des 10 nombres à répartir
compteur=0 #nombre de solutions
N1=[x for x in N if x!=0 ]#Liste pour le premier nombre à choisir
for i in N1:
    N2=[x for x in N if x!=i ]#Liste pour le deuxième nombre à choisir
    for j in N2:
        N3=[x for x in N2 if x!=j]
        for k in N3:
            N4=[x for x in N3 if x!=k]
            for l in N4:
                N5=[x for x in N4 if x!=l]
                for m in N5:
                    N6=[x for x in N5 if x!=m]
                    for n in N6:
                        N7=[x for x in N6 if x!=n]
                        for o in N7:
                            N8=[x for x in N7 if x!=o]
                            for p in N8:
                                N9=[x for x in N8 if x!=p]
                                for q in N9:
                                    N10=[x for x in N9 if x!=q]
                                    for r in N10:
                                        nombre1=100*i+10*j+k
                                        nombre2=100*l+10*m+n
                                        nombre3=1000*o+100*p+10*q+r
                                        if o!=0 and nombre1+nombre2==nombre3 and
nombre1<nombre2:
                                            compteur=compteur+1
                                            print(nombre1,nombre2,nombre3)

print(compteur)

```

**SOLUTION DÉFI N°143 – 2****« DES FILMS ACCÉLÉRÉS POUR LA TÉLÉVISION »**

*Un film n'a pas la même durée au cinéma et à la télévision car les films sont à l'origine tournés en 24 images par seconde, et projetés au même rythme dans les salles de cinéma. Sur nos téléviseurs, en raison de contraintes technologiques, les films sont en revanche diffusés en 25 images par seconde. Et donc très légèrement accélérés.*

*À l'œil, le changement est imperceptible. Mais à l'oreille on peut entendre une légère différence, notamment des voix plus aiguës, surtout lorsqu'il s'agit de films que l'on connaît bien, voire par cœur dans le cas des fans de Harry Potter.*

Question pour nos élèves : quelle est la différence de durée entre la diffusion du film « [Titanic](#) » au cinéma et une diffusion à la télévision française ?

Wikipédia fournit la durée du film : 188 minutes soit 188x60 secondes (11 280 secondes).

Pour 24 images par seconde, cela fait 270 720 images (11 280x24 = 270 720).

270 720 images à 25 images par seconde, cela fait 10 828,8 secondes soit 180min 28,8s (270 720:25 = 10 828,8).

Il reste à calculer « 188min-180min 28,8s ».

[Retour au sommaire](#)

## SOLUTION ALGO-RALLYE 143

Le défi algorithmique du PV 143 avait déjà été donné dans le PV 136. Toutefois, le défi posé était légèrement différent puisqu'il s'agissait de généraliser l'algorithme à des heures quelconques. Ainsi, on demandait de trouver la probabilité que le commissaire Girard, lors d'un réveil en sursaut entre 22h30 et 7h30, lise sur l'horloge une heure qui soit un carré parfait. La fonction `proba_carre_parfait` permet renvoie le nombre de jours cherchés.

Effectuer `proba_carre_parfait(22,30,7,30)` permet d'obtenir la réponse cherchée.

### Pseudo-code :

```

Fonction proba_carre_parfait(hc,mc,hl,ml : entiers ; num, den : entiers)
  nbCarres ← 0 ;      nombre de carrés obtenus
  h ← hc ; heure courante
  n ← mc ; minute courante
  total ← total_minutes(hc,mc,hl,ml) ;      la fonction total_minutes(hc,mc,hl,ml) le
                                             nombre total de minutes entre hc:mc et hl:ml

  tant que h ≠ hl , faire :
    si carre_parfait(100h+m) alors :      la fonction carre_parfait(n) renvoie
                                           vrai si n est un carré parfait
      nbCarres ← nbCarres + 1 ;
    finSi ;
    m ← m + 1 ;
    si m=60, alors :      on change d'heure
      h ← h+1 ;
      m ← 0 ;
      si h=24 alors :      on change de jour
        h ← 0 ;
      finSi
    finSi
  finTantque ;
  tant que m ≠ ml, faire :      on ajoute les minutes restantes
    si carre_parfait(100h+m) alors :
      nbCarres ← nbCarres + 1 ;
    finSi ;
    m ← m + 1 ;
  finTantque ;
  d ← pgcd(nbCarres,total) ;      réduction de la fraction
  num ← nbCarres/d ;
  den ← total/d ;
  renvoyer num, den.

```

---

### Python

```

from math import gcd

def total_minutes(hc,mc,hl,ml):
    """ Fonction total_minutes(hc,mc,hl,ml : entiers ; totalMinutes : entier)
    renvoie le nombre total de minutes, augmenté de 1, entre hc:mc et hl:ml
    """
    if hl<hc:      # si hl<hc, on n'est passé par 23:59
        totalMinutes=((24-hc)+hl)*60
    else:
        totalMinutes=(hl-hc)*60
    totalMinutes=totalMinutes+ml-mc+1
    return totalMinutes

```

```

def carre_parfait(n):
    """ Fonction carre_parfait(n : entier; booléen)
    renvoie Vrai si n est un carré parfait, Faux sinon
    """
    i=0
    while i**2<n+1:
        if i**2==n:
            return True
        i=i+1
    return False

def proba_carre_parfait(hc,mc,hl,ml):
    """ Fonction proba_carre_parfait(hc,mc,hl,ml : entiers ; den, num : entiers)
    renvoie le numérateur et le dénominateur de la probabilité de tomber sur un
    carré parfait entre les heures hc:mc et hl:ml
    """
    nbCarres=0
    h=hc
    m=mc
    total=total_minutes(hc,mc,hl,ml)
    while h!=hl:
        if carre_parfait(h*100+m):
            nbCarres+=1
            print(h,":",m)
        m+=1
        if m==60:
            h+=1
            m=0
            if h==24:
                h=0
    while m!=ml:
        if carre_parfait(h*100+m):
            nbCarres+=1
            print(h,":",m)
        m+=1
    d=gcd(nbCarres,total)
    num=nbCarres//d
    den=total//d
    return num,den

```

**PROBLÈME****LE PROBLÈME DU TRIMESTRE - N°144***Proposé par Philippe Févotte*

Soit une suite  $(u_n)$  définie pour les entiers  $n \geq 0$  et à termes positifs.

On considère la suite  $(v_n)$  définie pour tout entier par :

$$v_n = \sqrt{u_0 + \sqrt{u_1 + \sqrt{\dots + \sqrt{u_n}}}}$$

Donner une condition nécessaire et suffisante pour que la suite  $(v_n)$  converge

Le responsable de cette rubrique est [Philippe Févotte](#). Vous pouvez lui envoyer vos solutions à ce problème. **Des réponses, même partielles, seraient les bienvenues**, ainsi que toute proposition de nouveau problème.

[Retour au sommaire](#)