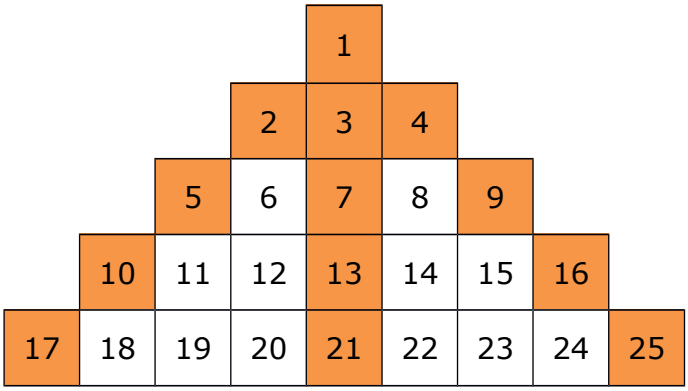
TROIS DÉFIS POUR NOS ÉLÈVES

Premier défi (n°131-a)



Voici une pyramide de nombres. Quels seront les nombres dans les cases oranges de la dixième ligne ? De la 2017ième ligne ? De la $n^{\text{ième}}$ ligne ?

Second défi (n°131-b)

Le concierge de l'immeuble où habite Cédric Villani a trois filles. Il s'adresse à Cédric en lui précisant que le produit de leurs âges est égal à 36 et que la somme de leurs âges est égale au numéro de l'immeuble qu'ils habitent. Cédric doit deviner leurs âges.

Après quelques instants de réflexion, Cédric se déclare incapable de deviner ces trois âges. Le concierge, tout heureux de le mettre en défaut, lui annonce qu'il va en parler à son ainée. À ces mots, Cédric Villani lui donne aussitôt l'âge des trois filles.

Et vous, avez-vous trouvé leurs âges ?

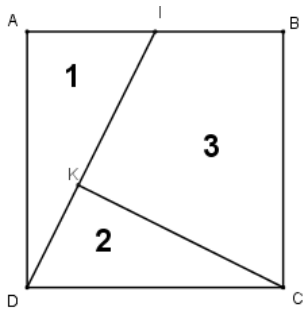
Troisième défi (n°131-c)

Dans la commune de Ville-la-Petite, il n'y a que 14 inscrits sur les listes électorales.
Au lendemain du référendum, qui posait la question « *Souhaitez-vous trouver un défi mathématique dans le bulletin municipal mensuel ?* », le journal local annonçait qu'il y avait eu 83,33 % de OUI.

Peux-tu dire combien de personnes ont voté OUI ?

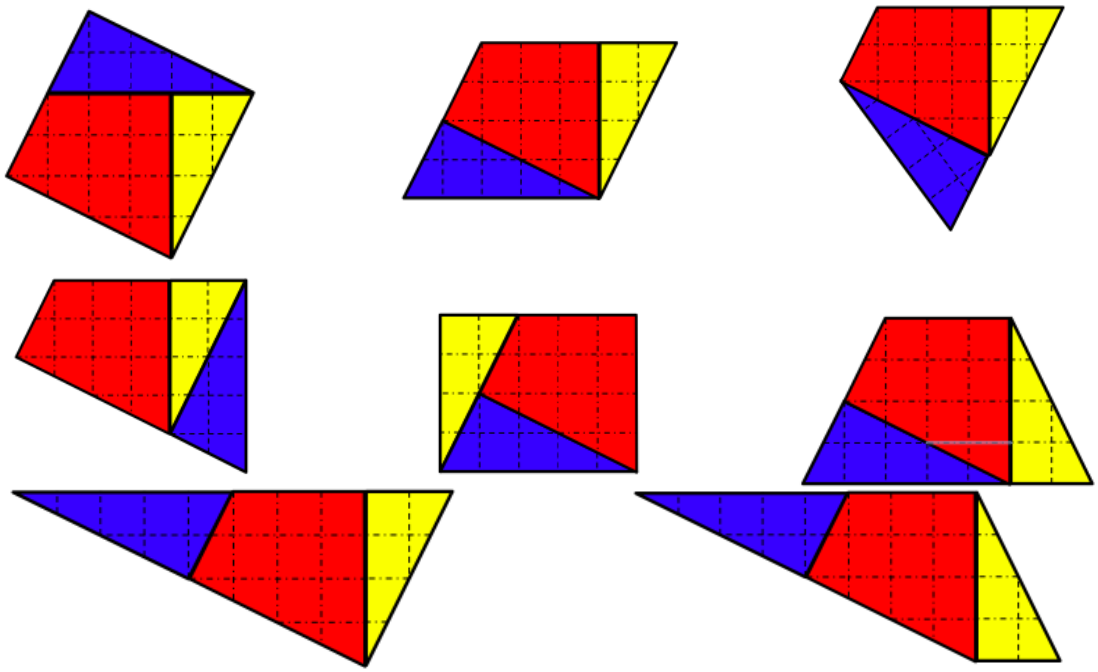
retour au sommaire

SOLUTION DU DÉFI « JEUNES ÉLÈVES » DU PV130



ABCD est un carré. I est le milieu du côté [AB]. La droite perpendiculaire à la droite (ID) passant par le point C coupe la droite (DI) en K. Le puzzle est constitué des trois pièces « 1 », « 2 » et « 3 ». En utilisant ces trois pièces, combien de quadrilatères différents peut-on obtenir ?

Les pièces coloriées et quadrillées utilisées ci-dessous sont celles évoquées en particulier dans le Petit Vert n°127.



Considérer qu'un triangle est un quadrilatère dont un côté a une longueur nulle est sans doute contestable. Cependant, cette configuration nous a permis de découvrir un troisième trapèze !

D'autres quadrilatères peuvent-ils être obtenus ?
Une manipulation exhaustive des pièces nous fait penser que la collection est maintenant complète. Les lecteurs du Petit Vert peuvent nous contredire !

retour au sommaire

SOLUTIONS DES DÉFIS DU n°130

Merci à tous ceux qui nous ont envoyé des propositions de solutions ou des algorithmes

Rappel du deuxième défi (130-a) : Le roi et ses sujets

Il était une fois un roi qui régnait sur un tout petit royaume : 177 sujets majeurs (que nous nommerons sujet 1, sujet 2, sujet 3, … sujet 177). Après avoir lu les œuvres de Platon, il décida de régner "démocratiquement" en s'entourant d'un conseil de 17 membres, tirés au sort.

Il procéda comme suit : il se plaça au centre de la cour de son château, et ses sujets se placèrent autour de lui en formant une ronde (dans l'ordre : sujet 1, sujet 2, … sujet 177).
Le roi pointa du doigt ses sujets un par un (en commenant par le sujet 1) en récitant cette comptine de 15 syllabes :
« **Ce/se/ra/toi/qui/siè/ge/ras/dans/mon/con/seil/au/cha/teau** ».
A la quinzième syllabe, le sujet visé quitte la ronde et vient se placer auprès du roi.
Et le roi continue sa ritournelle à partir du sujet suivant, jusqu'à ce qu'il ait recruté son conseil de 17 membres.

Le défi était le suivant : quel est le nom du dix-septième sujet "recruté" ?

Solutions de ce défi

Première proposition, d'André Stef. Il y a 177 personnes : 177 = 11x15 + 12, donc 11 recrutés avant de commencer un second tour (et 12 sujets n'ont pas encore été pris en compte dans la comptine : les « sujets » de 166 à 177). Après le 11ème barré, il « reste » à compter 6x15 = 90 sujets non barrés, soit 90-12 = 78 dans le second tour. Comme sont déjà barrés : 15, 30, 45, 60, 75, le 78ème est donc le sujet 83.

Le 17ème sujet recruté est donc le sujet 83.

De manière plus générale : dans la liste de *n* sujets, on peut, en barrant de *p* en *p*, chercher le *k*-ème sujet barré.

Voici alors une fonction SCILAB réalisant ce calcul.

```
function [chaine]=sortiech(n, p, k)
    //  longueur n , modulo p, element k barres*
    for i=1:n
        c(i)=i
    end;
    x=0;
    for i=1:k
        compte=0;
        while compte<>p
                x=x+1;
                if x>n then
                     x=x-n;
                     end;
                if c(x)==x then
                     compte=compte+1;
                      end;
                end;
        c(x)=0;
        end
    chaine=x;
    endfunction;
```

retour au sommaire

Voici un algorithme pour résoudre ce défi, ainsi que le programme Python correspondant

```
Variables :
n, entier (nombre de sujets)
k, entier (nombre de membres)
p, entier (nombre de syllabes)

Déclarations :
c: tableau d'entiers [de 1 à n]
i: entier ; compte : entier.

Saisir n, k et p.
Pour i allant de 1 à n+1 mettre i dans le
tableau c
(fin de la boucle pour)
Pour i allant de 1 à k+1, mettre compte à 0
  Tant que compte ≠ p faire :
    Si n ≥n remplacer x par x-n
    Si c[x]=0 remplacer par x+1
    Ajouter 1 à compte
    Ajouter 1 à x
    (fin de la boucle "tant que")
  Remplacer c[x-1] par 0
(fin de la boucle pour)

Fin de l'algorithme
```

Le programme en Python :

```
defsortiech(npk) : #longueur n, modulo p,
element k barres
c=[]
for i in range(1,n+1):
  c.append(i)
x=0
for i in range(1,k+1):
  compte=0
  while compte<p:
    print("compte",compte)
    print("x",x)
    if x>=n :
      x=x-n
    if c[x]==0 :
      x=x+1
    compte=compte+1
    x=x+1
  c[x-1]=0
  print(c)
return x
n=int(input("Nombre de sujets :"))
p=int(input("Nombre de syllabes :"))
k= int(input("Nombre de membres :"))
print("Le numéro du ",k,"-ième membre est ",
                                  sortiech(n,p,k)
```

Et enfin un programme en Algobox :

```
1   VARIABLES
2     n EST_DU_TYPE NOMBRE
3     p EST_DU_TYPE NOMBRE
4     k EST_DU_TYPE NOMBRE
5     x EST_DU_TYPE NOMBRE
6     c EST_DU_TYPE LISTE
7     i EST_DU_TYPE NOMBRE
8     compte EST_DU_TYPE NOMBRE
9   DEBUT_ALGORITHME
10    LIRE n
11    LIRE p
12    LIRE k
13    POUR i ALLANT_DE 1 A n
14      DEBUT_POUR
15      c[i] PREND_LA_VALEUR i
16      FIN_POUR
17    x PREND_LA_VALEUR 0
18    POUR i ALLANT_DE 1 A k
19      DEBUT_POUR
20      compte PREND_LA_VALEUR 0
21      TANT_QUE (compte!=p) FAIRE
22        DEBUT_TANT_QUE
23        x PREND_LA_VALEUR x+1
24        SI (x>n) ALORS
25          DEBUT_SI
26          x PREND_LA_VALEUR x-n
27          FIN_SI
28        SI (c[x]==x) ALORS
29          DEBUT_SI
30          compte PREND_LA_VALEUR compte+1
31          FIN_SI
32        FIN_TANT_QUE
33      c[x] PREND_LA_VALEUR 0
34      FIN_POUR
35    AFFICHER x
36  FIN_ALGORITHME
```

retour au sommaire

Solution du premier défi du PV130

Rappel de ce défi : Combien y a-t-il de zéros dans la liste des nombres entiers depuis 1 jusqu'à 2017 (inclus) ?

Une première solution consiste à compter « à la main » l'ensemble des zéros... mais en s'y prenant intelligemment (pas question d'écrire les 1017 nombres et compter leurs zéros !).
Nombres avec un seul zéro :

- au chiffre des unités (10, …, 90, puis 110, …, 990, puis 1101, …, 1901) soit 171 zéros
- au chiffre des dizaines (101, ...909 puis 1101, …, 1901) soit 162 zéros
- au chiffre des centaines (1011, …, 1099 puis 2011, …, 2017) soit 88 zéros
  Soit au total 421 zéros

Nombres avec deux zéros :

- au chiffre des dizaines et celui des unités (100, …, 900, puis 1100,…, 1900) soit 36 zéros
- au chiffre des centaines et celui des unités (1010, …, 1090, puis 2010) soit 20 zéros
- au chiffre des centaines et celui des dizaines (1001, …, 1009, 2001, …, 2009) soit 36 zéros
  Soit au total 92 zéros

Nombres avec trois zéros :

- dans le chiffre des centaines, des dizaines et des unité : 1000 et 2000, soit 6 zéros.

Au total, **519 zéros**

Une autre méthode serait d'utiliser l'informatique.

- Avec Scratch :

- avec python :

```
from math import*

def nombrezeros (n):
    puis=10
    compt=0
    quotient=0
    while n>=puis:
        for i in range(puis,n+1):
            reste=i%puis
            quotient=(10*reste)//puis
            if(quotient==0):
                compt=compt+1
        puis=puis*10
    return compt

print(nombrezeros(2017))
```

retour au sommaire

- avec Algobox :

```
1   VARIABLES
2     entier EST_DU_TYPE NOMBRE
3     nombreDeZero EST_DU_TYPE NOMBRE
4   DEBUT_ALGORITHME
5     //de 1 à 99, c'est évident
6     nombreDeZero PREND_LA_VALEUR 9
7     POUR entier ALLANT_DE 100 A 999
8       DEBUT_POUR
9       //on teste le chiffre des unités
10      SI (entier%10==0) ALORS
11        DEBUT_SI
12        nombreDeZero PREND_LA_VALEUR nombreDeZero+1
13        FIN_SI
14      //on teste le chiffre des dizaines
15      SI ((floor(entier/10))%10==0) ALORS
16        DEBUT_SI
17        nombreDeZero PREND_LA_VALEUR nombreDeZero+1
18        FIN_SI
19      FIN_POUR
20    POUR entier ALLANT_DE 1000 A 2017
21      DEBUT_POUR
22      SI (entier%10==0) ALORS
23        DEBUT_SI
24        nombreDeZero PREND_LA_VALEUR nombreDeZero+1
25        FIN_SI
26      SI ((floor(entier/10))%10==0) ALORS
27        DEBUT_SI
28        nombreDeZero PREND_LA_VALEUR nombreDeZero+1
29        FIN_SI
30      //on teste le chiffre des centaines
31      SI ((floor(entier/100))%10==0) ALORS
32        DEBUT_SI
33        nombreDeZero PREND_LA_VALEUR nombreDeZero+1
34        FIN_SI
35      FIN_POUR
36    AFFICHER nombreDeZero
37  FIN_ALGORITHME
```

- avec Scilab :

Principe : M a un zéro à la place k dans son écriture $n = a_i\ a_{(i-1)}\ ....\ a_1\ a_0$ en base 10 (attention : avec k compris entre 0 et i-1) si le reste r de la division euclidienne de M par 10^{k+1} est strictement inférieur à 10^k. Il reste alors à faire une double boucle et compter.
Application de cette fonction scilab à n = 2017 :

```
function [y]=nombredezeros(n)
    compt=0
    for i=1:n
        li=floor(log10(i))
        for j=1:li
            if (modulo(i,10^(j))<10^(j-1))
                    compt=compt+1
                        end
            end
    end
    y=compt
endfunction
```

On en trouve 519 également (heureusement !)

retour au sommaire