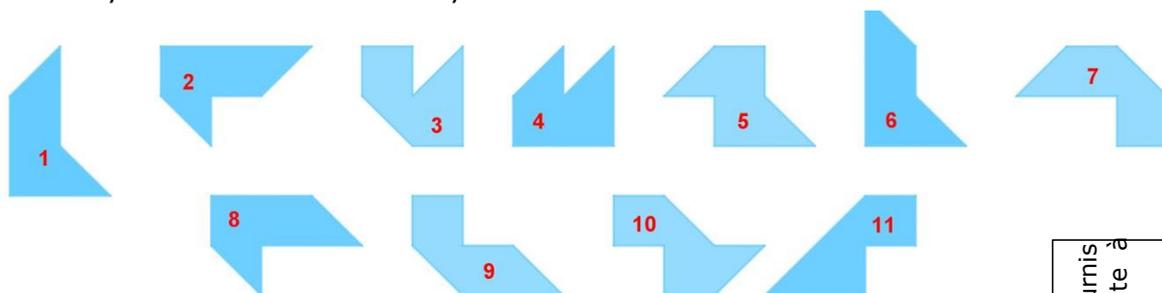


DÉFI N°138

En utilisant tous les **nombres** 1 ; 2 ; 3 ; 4 ; 5 et 6 au moins une fois, écrire une somme comportant le moins de termes possible et valant 2019.

SOLUTION DU DÉFI N°137 - LES BIMIS « L »

Pièces n'ayant ni axe ni centre de symétrie :



Pièces ayant un axe de symétrie :



Pièces ayant un centre de symétrie :



les dessins ont été fournis par Christine Oudin suite à un travail de groupe.

Remarque : les pièces n°17 et 19 admettent aussi deux axes de symétrie.

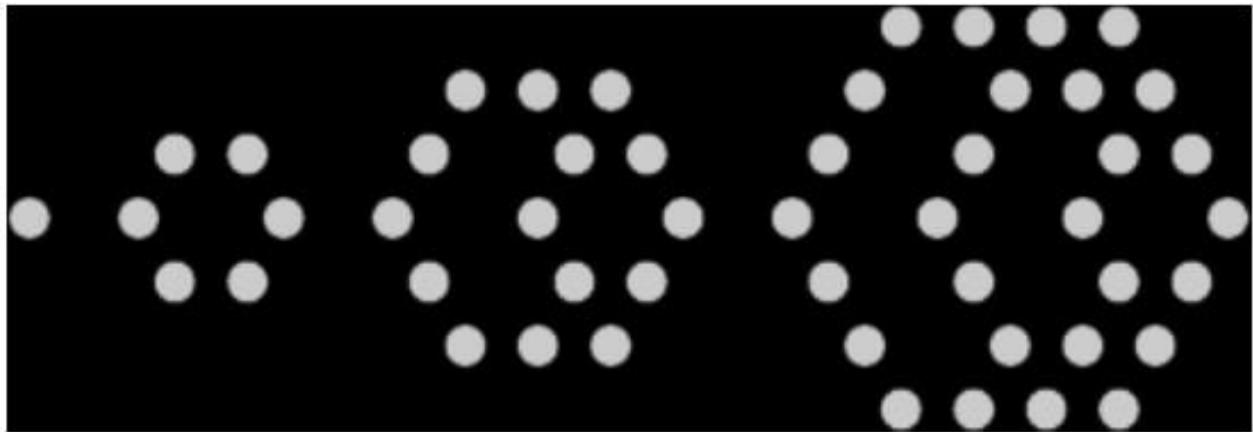
21 assemblages possibles de 2 trapèzes rectangles (composés d'un carré et d'un triangle rectangle isocèle) ont été trouvés.

DÉFI ALGORITHMIQUE N°138

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 ci-dessous avait été donné en 2016.

On dit que 2016 est une année hexagonale car il est possible de construire un hexagone régulier avec 2016 points



Les premiers nombres hexagonaux sont : 1, 6, 15, 28.

Sur le même principe, on peut construire la liste des nombres 24-gonaux, c'est-à-dire formant des polygones à 24 côtés. 2016 est aussi un nombre 24-gonal.

Quelle sera la prochaine année 24-gonale ?

Proposez une fonction qui, à partir d'un numéro d'année quelconque renvoie le numéro de la prochaine année 24-gonale.

SOLUTION DU DÉFI ALGORITHMIQUE N°137

Le défi algorithmique du PV 137 demandait de trouver l'heure à laquelle a retenti le 2019ème coup. La fonction `coup(n : entier ; h : entier)` renvoie le nombre d'heures `h` écoulées pour atteindre le `n`-ième coup. La fonction `date(heures : entier ; jour, mois, heure : entiers)` renvoie la date du jour et l'heure qui correspondent à un nombre heures d'heures écoulées depuis le premier janvier de l'année. Pour des raisons de lisibilité, la fonction ne tient pas compte des années bissextiles.

Effectuer `date_coup(2019)` permet d'obtenir la réponse cherchée.

Pseudo-code :

Fonction `coup(N : entier ; h : entier)`

`coup` ← 0 ;

`h` ← -1 ; (*la boucle commence avec h=0*)

tant que `coup < N`, **faire** :

`h` ← `h+1` ;

si `h` est multiple de 12, **alors** :

`coup` ← `coup + 12` ;

sinon :

`coup` ← `coup + reste(h,12)` ;

finSi ;

renvoyer `h`.

Fonction `date(heures : entier ; jour, mois, heure : entiers)`

`jours` ← `quotient(h,24)` ;

`heure` ← `reste(heures,24)` ;

`joursMois` ← [31,28,31,30,31,30,31,31,30,31,30,31] ; (*on suppose que la numérotation de la liste commence à 1*)

`mois` ← 1 ;

`nbJours` ← 31 ;

`jour` ← `jours` ;

tant que `nbJours < jours`, **faire** :

`jour` ← `jour - joursMois[mois]` ;

`mois` ← `mois + 1` ;

`nbJours` ← `nbJours + joursMois[mois]` ;

finTantque ;

`jour` ← `jour + 1` ;

renvoyer `jour, mois, heure`

Fonction `date_coup(N : entier ; triplet d'entiers)`

renvoyer `date(coup(N))` ;

Code Python :

```
def coup (N) :
```

```
    """ Fonction coup(N : entier; h : entier)
```

```
    N : nombre total de coups de l'horloge
```

```
    coup : entier, nombre courant de coups
```

```
    renvoie h, le nombre total d'heures écoulées au N-ième coup
```

```
    """
```

```
coup=0
h=-1
while coup<N:
    h=h+1
    if h%12==0:
        coup=coup+12
    else:
        coup=coup+h%12
return h
```

```
def date(heures):
```

```
    """ Fonction coup(heures : entier; jour, mois, heure :
entiers)
    heures : nombre d'heures écoulées depuis le premier janvier
    jours : entier, nombre de jours correspondant à heures
    heure : entier, nombre d'heures restant après jours
    joursMois : tableau d'entiers, liste des nombres de jours par
mois de l'année
    mois : entier, numéro du mois de l'année
    nbJours : entier, nombre courant de jours
    jour : entier, nombre de jours restant
    renvoie jour, mois et heures correspondant à heures
    """
```

```
    jours=heures//24
    heure=heures%24
    joursMois={1:31,2:28,3:31,4:30,5:31,6:30,7:31,8:31,9:30,
    10:31,11:30,12:31}
    mois=1
    nbJours=31
    jour=jours
    while nbJours<jours:
        jour=jour-joursMois[mois]
        mois=mois+1
        nbJours=nbJours+joursMois[mois]
    jour=jour+1
    return jour,mois,heure
```

```
def date_coup(N):
```

```
    """ Fonction date_coup(N : entier; triplet d'entiers)
N : nombre de coups de l'horloge
renvoie la date et l'heure correspondant à N
    """
    return date(coup(N))
```