

# BATTLE CODE

15 mars 2018 - ORLEANS

# MENU DU JOUR

- Les sujets sont indépendants les uns des autres
- Chaque équipe peut les traiter dans l'ordre qu'elle souhaite, et est libre de se les répartir à sa guise
- Chaque sujet doit aboutir à un programme qui sera mis à l'épreuve par l'arbitre par la fourniture de données d'entrée. Un des membres de l'équipe devra entrer ces données dans le programme, le faire tourner et montrer le résultat, qui sera vérifié par l'arbitre. Il y aura généralement plusieurs jeux de données successifs à entrer.
- Si toutes les réponses sont bonnes, les points du sujet sont attribués.
- Si de plus l'équipe est la première à résoudre le sujet, les points bonus lui sont aussi attribués
- Si en revanche une réponse n'est pas correcte, les données suivantes ne sont pas fournies et les points malus sont infligés.
- L'équipe peut soumettre par la suite une version améliorée de son programme (la procédure ci-dessus s'applique alors, en repartant du premier jeu de données)
- Le langage de programmation utilisé est libre et peut varier d'un sujet à l'autre pour une même équipe

Sujet	Points	Bonus	Malus
Les bières du développeur	6	1	1
Des Mois et des Jours	7	1	1
Audioactivité	7	1	2
Kamikaze	8	2	1
Premier Facteur	10	2	1
L'Alignement des Planètes	12	3	2
Ya les bons codeurs...	14	2	1
Fléchettes	15	3	2
Pierre Papier Ciseaux	16	3	2
Cavaliers en cavale	16	3	3
Des Chiffres et pas de lettres	20	4	2
Diviser pour régner	25	5	2



# LES BIÈRES DU DÉVELOPPEUR

---

- Introduction

- Un bon développeur finit toujours sa bière !!

- Règles

- La réserve de bières du pub contient X litres de bière
  - Le développeur commence à les boires en girafes de 3L...
  - Puis en pinte de 0,5L...
  - Puis en demi de 0,25L...
  - Et enfin en galopin de 0,125L
  - Sachant que chaque verre doit être vide avant d'en commander un autre
  - **Combien de consommation de chaque type prendra-t-il ?**

- Le programme doit

- Prendre en entrée le nombre de litres dans la réserve du pub au début de la journée de travail
  - Afficher en sortie le nombre de girafes, de pintes, de demis, de galopins
  - **Exemple** : 7,4 (litres de bière) → 2 (girafes), 2 (pintes), 1 (demi), 1 (galopin)



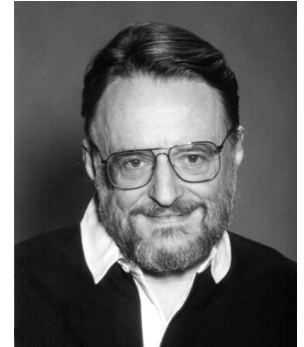
# DES MOIS ET DES JOURS

---

- Un client spécialisé dans la retraite a besoin de connaître, entre deux dates, le nombre de mois complets et le nombre de jours « hors mois complets », afin de pouvoir calculer correctement les cotisations de ses adhérents.
- Exemple : pour l'intervalle [29/01/2016, 05/04/2016] il compte :
  - Deux mois complets : février et mars
  - 8 jours hors mois complets :  
les 29, 30 et 31 janvier ainsi que les 1er, 2, 3, 4 et 5 avril
- Votre programme devra accepter en entrée deux dates au format dd/mm/yyyy, et produire en sortie le nombre de mois et le nombre de jours, sachant que :
  - La seconde date sera supérieure ou égale à la première
  - Les jours correspondant aux dates fournies font « partie » des jours à compter dans le calcul

3j 2m 5j





# AUDIOACTIVITÉ

- Voici une suite mathématique inventée en 1986 dite « audioactive ».
- Dans cette suite, un terme se détermine en annonçant les chiffres formant le terme précédent.

```

0  11
1  21 (sur la ligne précédente il y a 2 fois le chiffre « un »)
2  1211 (sur la ligne précédente il y a 1 fois le chiffre « deux » puis 1 « un »)
3  111221
4  312211 (sur la ligne précédente il y a 3 « un » puis 2 « deux » puis 1 « un »)
5  13112221

```

- Vous devez créer un programme capable, à partir d'un nombre initial positif choisi par l'utilisateur, de fournir l'itération  $n$  (choisie également par l'utilisateur) pour cette suite audioactive.
- Exemple:

Quel nombre initial ? 5

Quel rang ? 8

La suite audioactive de rang 8 est : 1113122113121113222115



# KAMIKAZE... OU PAS

- Introduction

- Pendant la seconde guerre mondiale, les kamikazes japonais étaient réputés pour leur sens du dévouement, n'hésitant pas à se sacrifier pour favoriser la victoire
- Un kamikaze se retrouva avec plusieurs de ses compagnons dans une grotte d'Iwo Jima alors que la défaite de cette bataille était inéluctable. Afin de sauver l'honneur et n'ayant plus qu'un seul sabre, ils décidèrent de se tuer avec, l'un après l'autre, pour ne pas tomber aux mains des américains.
- Ce kamikaze n'étant finalement pas très motivé pour mourir, mais n'osant s'opposer au groupe, il proposa que l'on se mette en cercle et que chaque troisième personne se suicide, en commençant par la première de la ronde.

- Problème

- Pour un groupe de  $N$  kamikazes ( $N$  entier strictement positif), quelle place doit-il choisir pour être le dernier à se suicider (et donc... à ne pas le faire 😊)

- Exemple

- pour  $N = 3$  il faut qu'il choisisse la 3<sup>e</sup> place :



# PREMIER FACTEUR

---

- Un nombre premier est un nombre entier naturel qui n'admet que deux diviseurs distincts entiers positifs (1 et lui-même).
  - 2, 3, 5 sont premiers.
- Deux nombres sont premiers entre eux lorsque le Plus Grand Commun Diviseur (PGCD) qu'ils partagent est 1.
- Réalisez un programme qui, à partir de deux nombres entiers positifs saisis en entrée, indiquera si ils sont premiers entre eux. De plus, le programme devra fournir la décomposition en facteurs premiers de chacun des nombres en entrée, en prenant soin de classer ces facteurs du plus grand au plus petit.
- Exemple 1 :

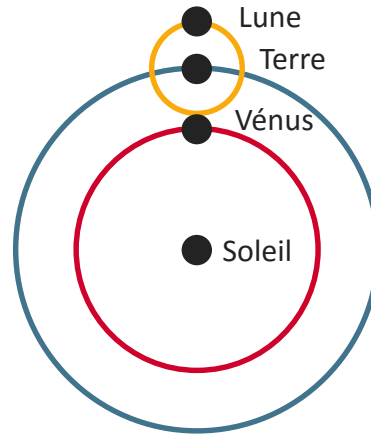
```
Entrez le premier nombre : 45
Entre le deuxieme nombre : 21
45 = 5*3*3
21 = 7*3
Ces nombres NE sont PAS premiers entre eux ! Leur PGCD est : 3
```
- Exemple 2 :

```
Entrez le premier nombre : 5
Entre le deuxieme nombre : 7
5 est premier
7 est premier
Ces nombres sont premiers entre eux !
```



# ALIGNEMENT ASTRAL

- La Terre et Vénus tournent autour du Soleil. La Lune tourne autour de la Terre. A une date donnée, l'alignement Soleil/Terre/Vénus/Lune est parfait et respecte le schéma suivant :



- On prendra les périodes de révolution suivantes :  
Terre : 365 jours | Vénus : 244 jours | Lune : 28 jours
- On partira du principe de le temps n'est pas continu mais quantifié et que l'unité de mesure minimale est le jour (pas d'alignement possible entre 2 jours).
- Nous sommes le 15/03/2018 et les astres sont alignés. Quelle sera la date du prochain alignement ?**
- Un dérèglement cosmique change la période de révolution de la Terre à une valeur  $X$  lors de ce nouvel alignement. **Pour un  $X$  donné, votre programme devra donner la date de l'alignement suivant.**



# YA LE BON CODEUR, ET YA LE MAUVAIS CODEUR...

## • Introduction

- Un client souhaite une IHM pour l'application qu'il va mettre prochainement sur le marché. Une équipe est en charge des développements. Dans l'équipe, il y a des bons, et des mauvais codeurs...

## • Règles

- Un bon développeur est capable de terminer une page en X jours
- Un mauvais développeur corrompt une page en Y jours, ce qui ralentit le bon développeur
- Chaque codeur, bon ou mauvais, développe 8h par jour
- Le but est de calculer en combien de temps les pages seront prêtes.
- Le résultat est attendu au format 00d01h20m30s, le résultat étant à tronquer à la seconde inférieure :
  - Si le résultat exact est 192,6666... secondes, la réponse attendue est **00d00h03m12s**
  - Si les mauvais codeurs empêchent les bons codeurs d'avancer, la réponse attendue est **FAIL**

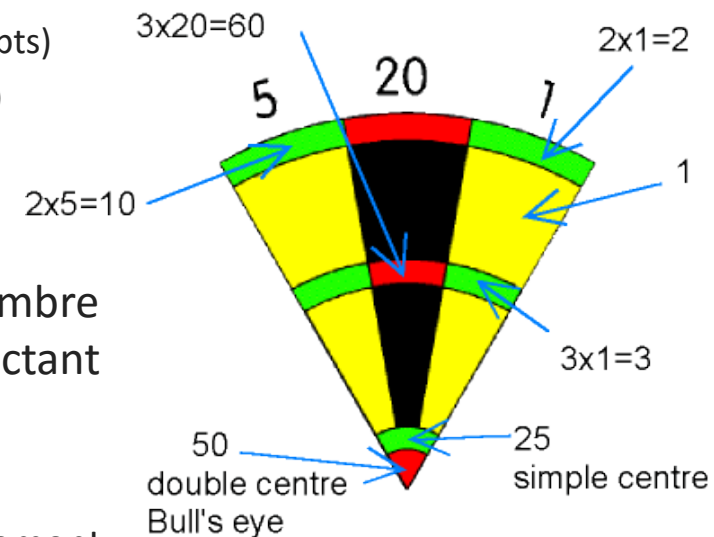
## • Exemple

- On doit faire 2 pages avec 1 bon codeur qui fait 1 page par jour et 2 mauvais codeurs qui corrompent chacun 1 page en 4 jours. Le programme est alimenté avec la séquence de nombres suivantes : 2 1 -4 -4
- Il faut 4 jours pleins pour que les 2 pages soient terminées (au bout de 2j, le bon dev a produit les 2 pages, mais les 2 mauvais dev ont corrompu chacun une demi page, il n'y a donc qu'une page terminée. Il faut donc le double soit 4j pour avoir les 2 pages terminées).  
Le résultat attendu est **04d00h00m00s**



# FLÉCHETTES – 501 INVERSÉ

- Un cible de fléchette est composée de la manière suivante
  - Un centre qui vaut 50 pts
  - Un bulle autour du centre qui vaut 25 points
  - 20 secteurs numérotés de 1 à 20 divisés en 4 parties :
    - 2 parties simple valant le numéro (1=1pt, 15=15pts)
    - 1 partie double valant le double du numéro (1=2pts, 15=30pts)
    - 1 partie triple valant le triple du numéro (1=3pts, 15=45pts)
- A partir d'un score en entrée, donner le plus grand nombre de coups nécessaires pour atteindre ce score en respectant les règles suivantes :
  - Chaque case ne peut être utilisée qu'une seule fois
  - Le nombre de points marqués au tour N doit être strictement supérieur aux nombre de points marqué au tour N-1



# PIERRE-PAPIER-CISEAUX

---

- Le principe de ce jeu est simple. Au même instant, deux adversaires présentent l'un des trois signes suivants : La pierre, le papier, les ciseaux. Chaque signe remporte la victoire sur un autre signe et perd contre un autre selon la règle :
  - Pierre gagne contre Ciseaux ( $P_i > C$ )
  - Ciseaux gagne contre Papier ( $C > P_a$ )
  - Papier gagne contre Pierre ( $P_a > P_i$ )
  - En cas d'égalité, la partie est nulle (aucun point n'est marqué).
- Deux joueurs A et B s'affrontent sur 100 parties successives à Pierre-Papier-Ciseaux, mais choisissent secrètement de suivre une stratégie prédéterminée :
  - Le joueur A choisit de répéter la séquence suivante:  $P_i, C, P_i, P_a, P_a, P_i$ .
  - Le joueur B choisit une stratégie "adaptative" qui dépend de la partie précédente :
    - si A joue  $P_i$  en à la partie  $n-1$ , alors B jouera  $P_a$  en partie  $n$
    - si A joue  $C$  en à la partie  $n-1$ , alors B jouera  $P_i$  en partie  $n$
    - si A joue  $P_a$  en à la partie  $n-1$ , alors B jouera  $C$  en partie  $n$
    - B joue  $P_a$  à la première partie
- Questions
  - Si chaque victoire donne 1 point au joueur vainqueur, qui aura le plus haut score au bout des 100 parties ?
  - Quels seront les scores ?



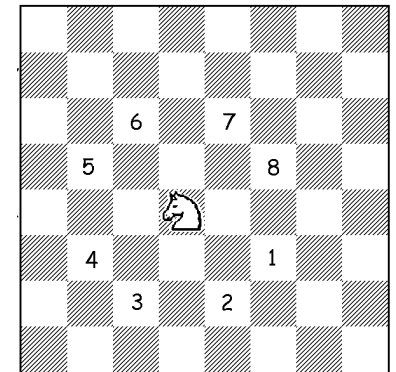
# CAVALIERS EN CAVALE

## • Introduction

- Un cavalier s'est échappé ce midi de son échiquier et se promène librement. Il est blanc, a une crinière plutôt longue, les yeux marrons, une haleine de cheval, se déplace en « L », a été vu pour la dernière fois dans le bar du foin « en haut à gauche ».

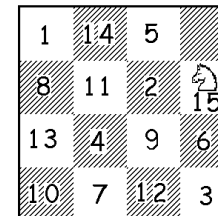
## • Règles

- Le cavalier est sur un échiquier de taille NxM (fourni dans la question)
- Une case est un bar à foin.
- Il doit changer de bar à foin toutes les 15 minutes s'il ne veut pas se faire repérer.
- Le cavalier commence son parcours dans le coin en haut à gauche (départ de sa cavale).
- Le cavalier se déplace en L dans un des 8 bars possibles (voir schéma)
- Le cavalier va toujours préférer aller en « 1 ».  
Si ce n'est pas possible, il va essayer d'aller en 2.  
Si ce n'est toujours pas possible, il va essayer d'aller en 3, etc...
- Le cavalier ne peut pas retourner dans un bar à foin qu'il a déjà visité (il se ferait repérer)
- La cavale s'achève quand le cavalier n'a plus de bar à foin directement atteignable. Il se fait alors arrêter.
- La réponse attendue est le temps (en minutes) de cavale du cavalier (première et dernière case incluses).



## • Exemple

- Sur un plateau 4x4, le cavalier peut aller dans 15 bars à foins différents. Sa cavale a donc duré 225 minutes. La réponse attendue est : **225**



# DES CHIFFRES ET...

- Tout le monde connaît l'(in)démorable jeu des chiffres et des lettres et son éternel « le compte est bon ».
- Le but de l'algorithme est, à partir d'une liste de 6 nombres entiers positifs et d'un résultat entier positif, de déterminer le meilleur enchainement d'opérations mathématiques (+, -, \* et / entière) permettant de trouver le résultat attendu, le tout en respectant les règles suivantes :
  - Après chaque opération, les nombres utilisés ne sont plus disponibles pour les autres calculs, mais le résultat de l'opération faite si.
  - La division ne peut être utilisée que si le résultat de la division entière a un reste nul.
  - La soustraction ne peut pas être utilisée si elle a pour conséquence l'obtention d'un nombre négatif (ie : attention à l'ordre des opérandes)
  - On considère comme meilleur l'enchainement du plus petit nombre d'opération donnant le résultat le plus proche de l'attendu,

## Exemple 1 :

Entrez les 6 nombres a utiliser : 1 5 10 50 9 4

Entrez le nombre a trouver : 742

Le compte est bon :

$$50 - 1 = 49$$

$$49 + 4 = 53$$

$$5 + 9 = 14$$

$$53 * 14 = 742$$

## Exemple 2 :

Entrez les 6 nombres a utiliser : 2 2 3 3 4 5

Entrez le nombre a trouver : 742

Je n'arrive qu'a 720 :

$$2 + 2 = 4$$

$$4 * 3 = 12$$

$$12 * 3 = 36$$

$$36 * 4 = 144$$

$$144 * 5 = 720$$



# DIVISER POUR MIEUX RÉGNER

- Remplaçons quelques instants les calculatrices par du papier et un crayon, et rappelons nous comment diviser.
- Réaliser un programme qui, à partir de deux nombre entiers positifs (a et b) choisis par l'utilisateur affiche à l'écran la division de a par b telle qu'elle serait posée par un enfant de primaire. Le programme devra proposer de réaliser une division euclidienne (entière) ou une division « normale ».

## Exemple de division euclidienne:

```
Entrez le numerateur : 237
Entrez le denominateur : 6
Type de division (1-entiere; 2-normale) : 1
237| 6
18 |---
---| 39
 57|
 54|
  --|
   3|

237 divise par 6 vaut 39, reste 3
```

## Même exemple pour une division « normale » :

```
Entrez le numerateur : 237
Entrez le denominateur : 6
Type de division (1-entiere; 2-normale) : 2
237 | 6
18  |-----
--- | 39,5
 57 |
 54 |
  --|
   30|
   30|
   --|
    0|

237 divise par 6 vaut 39,5
```

