

# Note

---

Émetteur(s) : Sopra Stéria Battle Team 2016

Destinataire(s) : Collaborateurs Sopra Stéria

Copie(s) :

Objet : **Exercices Battle Code 2016**

---

## 0. Introduction

---

Bienvenue à la Battle Code Group 2016 de Sopra Steria.

Les sujets exposés ci-après peuvent être réalisés dans n'importe quel ordre et ne sont pas dépendants les uns des autres. Charge à vous de trouver des solutions pertinentes et robustes permettant de résoudre les problèmes donnés.

Chaque exercice est composé de 3 questions (avec des jeux de données de plus en plus complexes). Théoriquement, les 3 questions ne nécessitent pas de modification de l'algorithme. Si ce dernier est bien écrit, la question 3 devrait être une formalité.

Même si le format attendu est clairement précisé dans chaque énoncé, il est rappelé que toutes les réponses demandées sont au format international (par exemple : 01d02h15m45s au lieu de 01j02h15m45s)

La question 3 contient généralement un grand jeu de données pour tester la fiabilité des algorithmes, elle peut alors être contenue dans un fichier texte. Il n'est pas nécessaire que le programme lise ce fichier. Un simple copier-coller du contenu dans une variable suffit.

**Soyez vigilants, l'arbitre virtuel ne réfléchit pas, donc parlez lui au format qu'il attend !**

Il n'y a qu'une seule bonne réponse par question. Si la réponse est une liste, elle doit donc être ordonnée comme précisé dans l'énoncé.

Bien lire les règles de chaque exercice pour donner la réponse dans le format attendu. Avant de répondre, pensez à relire l'énoncé pour être sûr du format ainsi que des conventions adoptées (sens des axes X et Y, indice de la première cellule...) !

Dans les exemples, les réponses attendues (c.-à-d. au bon format) sont en **vert**.

Les sauts de lignes (\n et \r) seront automatiquement supprimés des réponses envoyées à l'arbitre virtuel. Ils sont donc optionnels (exercices 7 et 16).

**Attention lors de l'envoi du code source. Il doit être auto-suffisant, il est destiné à l'équipe de validation. Le code doit être exécutable sur une autre machine. Fournissez les éventuelles librairies personnalisées.**

N'hésitez pas à fournir un fichier readme.txt avec vos sources pour indiquer à l'équipe de validation comment lancer votre code. Vérifiez bien que vous envoyez le bon code source.

L'équipe de validation ne testera que les 3 questions fournies dans l'énoncé.

Si le code source testé par l'équipe de validation ne permet pas de résoudre l'énoncé, les points de l'énoncé seront annulés.

Les exercices les plus faciles sont estimés à 10-15 minutes. Les exercices les plus difficiles sont estimés à 45-60 minutes.

Ces temps sont donnés à titre indicatif, l'équipe Battle Code n'est pas responsable de la performance des développeurs(euses).

À vous de vous définir votre stratégie ! Plus l'exercice est difficile, plus il est rentable, mais en cas d'erreur, le débogage est nettement plus chronophage.

Passons aux choses sérieuses ! Voici la liste des énoncés, triés par ordre croissant de difficulté.

Un bonus est attribué à la première équipe (mondiale) qui résout l'exercice.

Un malus est appliqué pour chaque mauvaise réponse (effet cumulatif).

Pour marquer les points, il faut répondre correctement aux 3 questions, et fournir le code source.

Numéro	Exercice	Points	Bonus	Malus
1	Match de foot	5	2	1
2	Message codé 1	6	2	1
3	Boby Lapointe	6	2	1
4	Comptage de M&M's	7	2	1
5	Message codé 2	7	2	1
6	Somme d'une horloge	7	2	1
7	Générateur de démineur	8	3	1
8	Message codé 3	10	3	1
9	Langage ixaire	10	3	1
10	House of cards	15	5	1
11	Les bons et les mauvais codeurs	20	7	2
12	Cavalier en cavale	25	8	2
13	Murs d'un labyrinthe	25	8	2
14	Puissance 4	25	8	2
16	Factures impayées	35	12	2
15	Construction d'un Pick-up Légo automatisé	45	15	2
17	Langage Battle Code Rocks	50	18	2

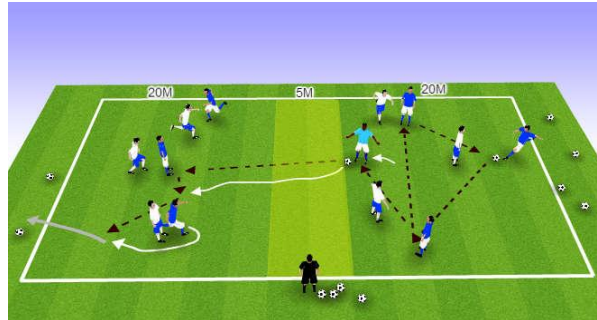
À vos claviers ! Prêts ? Codez !

# 1. Match de foot (5pts +2/-1)

---

## Introduction

Une équipe de football amatrice cherche à estimer ses chances de gagner en fonction de l'équipe qu'elle sélectionne. C'est un match de football classique. Pas un match de rugby, pas de football américain, ni de football gaélique ... C'est pas qu'ils ne sont pas intéressants, mais c'est plus compliqué. Le football classique, y'a un ballon, vous tirez, vous marquez, vous avez un point. C'est tout. N'importe quel enfant en est capable. Pour s'amuser. Gratuitement...



## Règles

- Notre équipe marque des buts contre l'adversaire toutes les X minutes de jeu. Le premier but est marqué à la X<sup>ième</sup> minute.
- Notre équipe marque des buts contre son camp toutes les Y minutes de jeu. Le premier but est marqué à la Y<sup>ième</sup> minute.
- L'équipe adverse est professionnelle, elle ne marque aucun but contre son camp.
- L'équipe adverse marque des buts toutes les Z minutes de jeu. Le premier but est marqué à la Z<sup>ième</sup> minute.
- La réponse attendue est le score final après 90 minutes de jeu.
  - Si on marque 3 buts dans les cages adverses, 1 but dans nos propres cages, et l'adversaire marque 3 buts dans nos cages, la réponse attendue est **3-4**

## Questions

Quel est le score entre ces équipes :

1. Notre équipe marque 1 but toutes les 35 minutes dans les cages adverses  
Notre équipe marque 1 but toutes les 65 minutes dans nos propres cages  
L'adversaire marque 1 but toutes les 35 minutes dans nos cages.
2. Notre équipe marque 1 but toutes les 15 minutes dans les cages adverses  
Notre équipe marque 1 but toutes les 30 minutes dans nos propres cages  
L'adversaire marque 1 but toutes les 10 minutes dans nos cages.
3. Notre équipe marque 1 but toutes les 1 minutes dans les cages adverses  
Notre équipe marque 1 but toutes les 120 minutes dans nos propres cages  
L'adversaire marque 1 but toutes les 2 minutes dans nos cages.

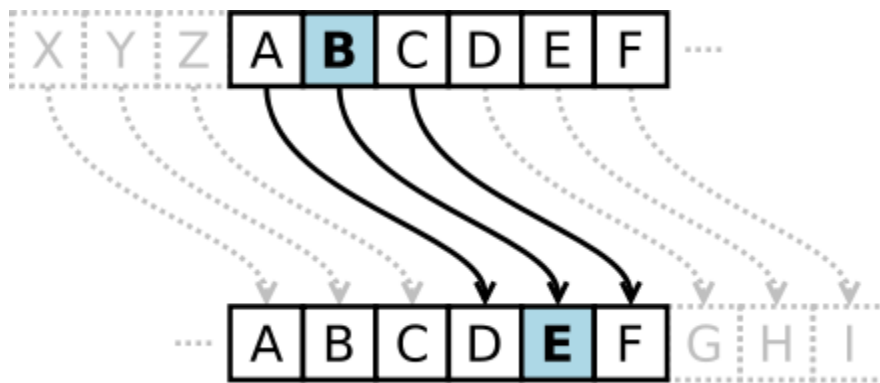
## 2. Message codé 1 (6pts +2/-1)

### Introduction

Une méthode très simple pour chiffrer un message est le chiffre de César. Il était utilisé il y a plus de 2000 ans garantissant ainsi que le message ne puisse être intercepté. Le principe est simple : il suffit de décaler les lettres de l'alphabet par un nombre fixe. Aujourd'hui, cette méthode n'est pas fiable car très connue, et il suffit de deviner une lettre (par analyse de fréquence) pour craquer le code, ou de tester les 25 possibilités jusqu'à ce qu'on lise un message intelligible.

### Règles

- Le chiffre de César est fourni dans la question
- Si le nombre de César vaut « +1 », alors A clair devient B chiffré



Dans l'image ci-dessus, le nombre de César vaut « +3 », donc « B » du message clair devient « E » dans le message codé.

- Le but de l'exercice est de déchiffrer un message codé.
- Les messages ne sont composés que des 26 lettres majuscules de l'alphabet latin.
  - Ils ne contiennent aucun accent.
  - Les espaces et caractères spéciaux ne sont pas codés

### Exemple

- Si le nombre de César est « +5 », le message « MJQQT BTWQI ! » devient « **HELLO WORLD !** »

### Questions

Déchiffrez les messages suivants :

1. Chiffre de César: +1. Message: DBFTBS DJQIFS
2. Chiffre de César: -3. Message: QEFP FP X KBDXQFSB HBV
3. Chiffre de César: +404. Message: TUZF: GEQ YAPGXA FIQZFK EUJ FA PQOAPQ FTUE AZQ !

### 3. Bobby Lapointe (6pts +2/-1)

#### Introduction

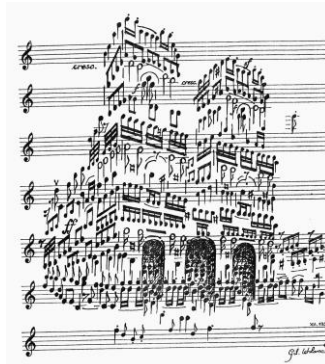
Le chanteur Bobby Lapointe (1922-1972) était aussi un chercheur et un informaticien avant l'heure. Il a inventé le système "bibli-binaire" qui utilise des syllabes au lieu des chiffres et des lettres pour coder des nombres en hexadécimal.

Le système ayant été breveté, on va le modifier pour utiliser la base 10, on l'appellera "dodo-deci". L'exercice consiste à écrire un convertisseur dodo-deci -> décimal.

#### Tables de correspondance

Chaque chiffre devient une syllabe, en utilisant la correspondance suivante :

décimal	dodo
0	po
1	pa
2	pe
3	pi
4	to
5	ta
6	te
7	ti
8	ko
9	ka



Afin d'utiliser le "dodo-déca", il ajouta également les opérateurs

opérateur	dodo
+	do
-	da
*	de

#### Règles

- Les priorités classiques d'opérations sont respectées, c'est-à-dire que la multiplication est prioritaire par rapport aux additions et soustractions.
  - $\text{padopedepidape} = 1 + 2 * 3 - 2 = 1 + (2 * 3) - 2 = 5$
- Le premier caractère fourni ne peut pas être un « d », c'est-à-dire que l'opération commence forcément par un nombre positif.
- Un opérateur est TOUJOURS entouré de deux chiffres. Il est impossible de retrouver deux opérateurs successifs.
  - Il est donc inutile de prévoir les cas « ++ », « +- », « -+ », « \*+ » et « \*- »
- Tous les nombres sont des entiers.

#### Exemple

Sans calcul	Question	pepopate
	Réponse	2016

Avec Calcul	Question	todopedepipedapapa
	Décimal	$4 + 2 * 32 - 11$
	Réponse	57

#### Questions

- papopeto
- papedotidepetadokadopatekadatatidopipidopetidetotidatitopidotepetadapipeta
- papedotokopadotidepetadapatadepekodepadekadopatekadotatipipidopetidetotidatitopidotepe  
tadotokadetadekodepodetotidapipetadopepadekopedopipokadakopopedatatopipidotitedopo

## 4. Comptage de M&M's (7pts +2/-1)

### Introduction

Quand on achète des M&M's, on doit être sûr de ce qu'on achète. Si nos préférés sont les bleus, il y a intérêt à ce qu'ils ne soient pas minoritaires ! Que dites-vous de le vérifier ensemble ?

### Règles

- Tout commence avec un paquet vide.
- L'entrée est une suite de « C NNN » séparés par des virgules.
  - « C » est la couleur (R, G, B, Y, O) de M&M's
  - « NNN » est le nombre de M&M's ajoutés (compris entre 000 et 999)
- Plusieurs lignes peuvent avoir la même couleur
- Le but est de donner la couleur de M&M's majoritaire.



### Exemple

Entrée : R 005,R 010,Y 090,G 020,B 010,R 008,Y 090,O 170

La réponse attendue est **Y**, car c'est la couleur de M&M's la plus présente (180 M&M's)

### Questions

Quelle est la couleur majoritaire dans ces paquets ?

1. O 018,Y 028,R 097,G 078,O 071,O 045,B 037,O 055,Y 005,R 054
2. R 204,B 948,Y 077,Y 937,O 247,O 478,O 285,Y 194,O 466,B 922,G 067,B 315,R 597,B 293,O 995,R 477,Y 716,R 965,G 770,R 227
3. B 828,R 688,R 867,R 810,R 720,R 236,R 223,R 900,G 544,G 312,G 879,B 616,B 646,B 898,R 155,R 255,G 296,G 469,Y 000,G 726,G 347,G 330,B 436,B 139,B 129,B 395,B 687,B 844,R 601,R 524,R 548,R 238,R 231,G 251,G 013,G 664,G 928,G 996,B 598,B 855,B 354,B 926,B 645,B 771,B 188,B 025,B 437,B 323,R 419,R 573,R 948,R 925,R 279,R 776,R 601,R 369,R 892,R 742,R 839,R 967,G 246,G 543,G 890,G 454,G 577,G 563,G 548,G 595,G 758,G 869,G 313,G 989,B 786,B 806,B 755,B 017,B 965,B 548,B 414,R 423,G 877,B 815,R 106,G 477,B 728,R 488,G 969,B 209,R 177,G 315,B 091,R 660,G 226,B 462,G 064,R 644,G 274,B 096,R 314,G 966,B 933,B 731,B 542,G 584,R 683,R 571,R 126,G 666,B 978,B 535,G 827,G 031,G 889,B 213,B 636,B 924,R 873,R 465,R 122,R 603,R 769,G 860,B 100,G 706,R 616,G 713,B 989,R 869,G 378,B 899,G 681,R 553,G 178,B 191,R 289,G 567,B 126,R 326,G 587,B 234,R 351,G 524,B 516,R 692,G 311,B 677,R 580,R 274,R 985,R 926,R 488,R 462,G 044,G 842,G 868,G 357,G 588,G 943,G 426,B 157,B 542,B 743,B 361,B 599,B 311,G 075,R 403,B 835,G 789,R 894,G 310,B 709,R 301,G 273,G 165,G 919,G 589,B 836,B 321,R 007,R 431,R 308,R 335,G 667,G 976,G 445,B 738,B 332,B 407,B 019,R 983,R 507,R 009,R 115,R 408,B 818,B 247,B 432,B 575,R 252,R 942

## 5. Message codé 2 (7pts +2/-1)



\*match = allumette

### Introduction

Le chiffre de César étant trop connu, inventons une nouvelle méthode de cryptage : Le décallage-alphabétique-qui-dépend-de-la-lettre-précédente-ce-qui-empêche-toute-analyse-frequencielle-et-qui-est-donc-impossible-à-déchiffrer-à-moins-d-en-connaître-la-méthode-d-encryptage. C'est parti !

### Règles

- La première lettre du message codé est en clair
- La lettre suivante est décalée d'autant que la valeur de la lettre codée précédente.
  - Si la première lettre est un A, la seconde sera décalée d'un cran
  - Si la 3<sup>ème</sup> lettre du message codé est « E », alors la 4<sup>ème</sup> lettre sera décalée de 5 crans
- Le but de l'exercice est de déchiffrer un message codé.
- Les messages ne sont composés que des 26 lettres majuscules de l'alphabet latin.
  - Ils ne contiennent aucun accent.
  - Les espaces et caractères spéciaux ne sont pas codés.

### Exemple

- Si l'on encode « **SOPRA STERIA!** » on obtient « SHXPQ JDIAJK! »
  - S est la 19<sup>ème</sup> lettre →  $19+0 = 19$  → « S »
  - O est la 15<sup>ème</sup> lettre →  $15+19 = 34$  →  $34-26 = 8$  → « H »
  - P est la 16<sup>ème</sup> lettre →  $16+8 = 24$  → « X »
  - R est la 18<sup>ème</sup> lettre →  $18+24 = 42$  →  $42-26 = 16$  → « P »
  - Etc...

### Questions

Déchiffrez les messages suivants :

1. HMEJ GL SH!
2. YNI BJKWI WLF VWPI...!!!
3. YXWLAPETOJEZU ... NVWIU IXMBQFZT ... JZABCDEFGHIJKDWPIB...



## 6. Somme d'une horloge (7pts +2/-1)

---

### Introduction

Prenons une horloge digitale qui affiche les heures et les minutes (format 24h). Que se passe-t-il si on fait la somme des heures et des minutes ?

### Règles

- La période ne peut pas dépasser 24h.
- A 08h53, la somme des heures et des minutes est **61**



### Exemple

- De 15h15(inclus) à 15h18 (non inclus), la somme des heures et des minutes est 30 (15h15) +31 (15h16) +32 (15h17) = **93**

### Questions

Quelle est la somme sur ces périodes ?

1. 00h00 à 01h00
2. 09h59 à 23h30
3. 15h00 à 14h58

## 7. Démineur (8pts +3/-1)

### Introduction

Tout le monde connaît le fameux démineur sur Windows, où le but est de trouver toutes les mines sans marcher dessus. Quand on marche sur une case, le nombre affiché est le nombre de mines qui entourent la case. Nous proposons ici de générer un plateau de démineur !

### Règles

- Le plateau est un carré de côté N
- Les données en entrée sont les coordonnées des mines
  - Un espace sépare les coordonnées X et Y de la mine
  - Une virgule sépare chaque mine
- La case en haut à gauche est la case (1,1)
- La réponse attendue est la description du plateau entier.
  - Un chiffre indiquant le nombre de mines voisines
  - Un « x » sur une mine
- Dans la réponse, les retours à la ligne seront tronqués

### Exemple

Sur le plateau 7x7 suivant :

4 1,6 1,2 2,5 3,7 3,2 6,4 6,6 6

La réponse attendue est:

```
112x2x1
1x22332
1111x2x
0001121
1121211
1x2x2x1
1121211
```



### Questions

Écrire un programme qui génère les plateaux suivants :

1. Plateau 5x5 : 5 5,2 3,1 1,3 3,2 5
2. 10x10 : 8 7,10 3,10 2,3 5,4 8,9 5,3 2,4 10,9 7,1 8,4 3,5 2,2 10,5 7,4 9,6 2,2 1,6 3,3 10,6 10
3. 100x100 : Minesweeper\_q3.txt

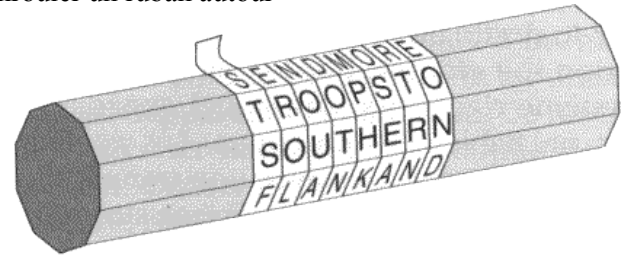
## 8. Message codé 3 (10pts +3/-1)

### Introduction

Chez les Spartiates, la scytale, également connue sous le nom de bâton de Plutarque, était un bâton de bois utilisé pour lire ou écrire une dépêche chiffrée. L'idée est d'enrouler un ruban autour d'un bâton afin de lire le message.

### Règles

- La première lettre du message codé est la première lettre du message final
- Dans la question, le diamètre du bâton à utiliser sera fourni, ainsi que la hauteur de chaque lettre
  - Le nombre de lettre par tour de bâton sera un entier. Si le résultat du calcul donne 9.023 lettres, il devra donc être arrondi à 9.
- La question est de la forme « XX,YY,MESSAGE »
  - XX est le diamètre du bâton en centimètres
  - YY est la hauteur d'une lettre en millimètres
- Le but de l'exercice est de déchiffrer un message codé.
- Les messages ne sont composés que des 26 lettres majuscules de l'alphabet latin.
  - Ils ne contiennent ni accent, ni espace, ni ponctuation.
- Le message recouvre entièrement le bâton, il n'y a pas de caractères blancs à ajouter.
  - S'il y a la place pour 10 caractères par tour de bâton, le message aura un multiple de 10 caractères.



### Exemple

- Si l'on décode « STSFEROLNOUADOTNMPHKOSEARTRNEOND » sur un bâton de d'1,4cm de diamètre, des lettres d'1cm de hauteur, on obtient 4 lettres par tour de bâton. Le message fait 32 caractères, ce qui implique 8 tours de bâton. On obtient « **SENDMORETROOPSTOSOUTHERNFLANKAND** ».

### Questions

Déchiffrez les messages suivants :

1. 13,102,LIUTUMRHKYFEEOAR
2. 28,110,HHHPLOLWMAEEFOAAMTAECADLMIIDINELMSRV TUNOWTSEYNSW
3. 18,51,SODRNDULTRTOVSADILEYAIPIETBNTXASORDRIUGRANTNAEVOSPANDETSSINIR  
NDPRETOCSNISCEICENEEOVFRRAHREOFSAOIFENIOFTSTRTONOAFWPEMIRALETEAR  
AACMBOUHRONTAALGREIECDILNEYOMNDEPOBCSYPOGESUNUEIEESSVSBPSSTSATI  
ESLRIESTNCNLEIONRCHLOTORCGEVLIEHPVORSIISAPEMIRASCEIDRMECGMCENSEEA  
NEAMHSTTRHRTSNEAASHIEKISISLDTENNENOSTLDOADSTFPAHEEMNIICRRTANDASGV  
OAAITGVKWIENSSOAEAEETPSTTNDLSLTRAOURESDCUHLRLURTROEEFTTTTCIOEMABT  
RFITADSBNEHAONUIESIDSENLRSLTNITTSISETIHINUHFOYMRVENNSSIOSSAUEIGOERR  
OTNSRRHVODMFEATSMIAFQAEMGEUOGTIUTNSEDCSHINEIDIMBCTQOFSOTNEYECUN  
OTNOTNL SOASRIPEETESMLOMORNGAAFPIPAN

## 9. Langage ixaire (10pts +3/-1)

### Introduction

Le binaire c'est bien, mais tout le monde sait le lire. Un cryptologue a donc inventé le langage *ixaire*.

### Règles

- Le message en entrée est écrit en caractères ASCII qu'il faut traduire en binaire sur 7 bits.
- Le message encodé en sortie est constitué de blocs de « x »
- Un bloc est séparé d'un autre bloc par un espace
- Deux blocs consécutifs servent à produire une série de bits de même valeur (que des 1 ou que des 0 en binaire) :
  - Premier bloc (entête) : il vaut toujours x ou xx. S'il vaut x la série suivante contient des 1, sinon elle contient des 0.
  - Deuxième bloc (bits): le nombre de x dans ce bloc correspond au nombre de bits dans la série

### Exemple

Prenons un exemple simple avec un message constitué d'un seul caractère : « C ». C majuscule en binaire vaut 01000011 sur 8 bits, donc 1000011 sur 7 bits, ce qui donne en ixaire :

- x x (la première série est composée d'un seul 1)
- xx xxxx (la deuxième série est composée de quatre 0)
- x xx (la troisième série est composée de deux 1)

C vaut donc : **x x xx xxxx x xx**

Deuxième exemple, nous voulons encoder le message « B C » (soit les 21 bits 1000010\_0100000\_1000011) :

- x x (un seul 1)
- xx xxxx (quatre 0)
- x x (un 1)
- xx xx (deux 0)
- x x (un 1)
- xx xxxxx (cinq 0)
- x x (un 1)
- xx xxxx (quatre 0)
- x xx (deux 1)

« B C » vaut donc : **x x xx xxxx x x xx xx x x xx xxxxx x x xx xxxx x xx**

### Questions

Écrire un programme qui écrit ces messages en ixaire :

1. BC
2. battle code rocks
3. Chuck N0rr1s' k3yb0@rd h@s 2 k3ys: x @nd wh1t3 sp@c3.



## 10. House of cards (15pts +5/-1)

### Introduction

Une Startup souhaite créer un mélangeur automatique de carte et a mis au point un prototype Alpha permettant de sortir une carte « aléatoire » d'un paquet. L'aléatoire n'existant pas vraiment, il se trouve qu'en connaissant les cartes du paquet en entrée, on peut prédire quelle carte sera sortie.

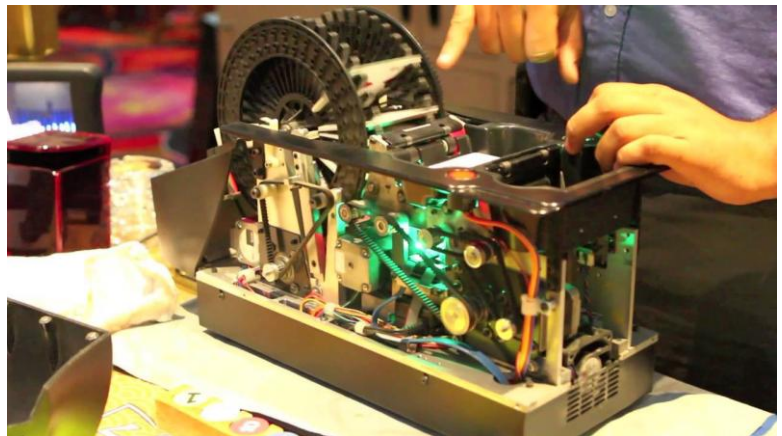
On se propose de deviner la carte de sortie, lorsqu'on fournit un paquet de cartes trié en entrée.

### Règles

- Les cartes suivent l'ordre de valeur du poker : AS, ROI, DAME, VALET, 10, 9, 8, 7, 6, 5, 4, 3, 2 ; et l'ordre des couleurs : PIQUE, CŒUR, CARREAU, et TREFLE.
- Ces cartes sont numérotées de 1 à N, avec N le nombre de cartes fournies. La carte n°1 étant sur le dessus, puis la carte n°2... jusqu'à la carte N. La première carte du tas est donc l'AS DE PIQUE et la dernière le 2 DE TREFLE.
- Le mélangeur prend la carte du haut (**n°1 = AS de PIQUE**), la replace au-dessous du paquet, jette la suivante (**n°2 = ROI DE PIQUE**), replace la n°3 au-dessous, jette la n°4 etc...
- La carte de sortie est la dernière carte restante.
- La réponse attendue est le numéro de la carte dans le tas initial, la couleur, ainsi que la valeur de la carte :
  - Valeurs : AKQJT98765432.
    - **Notez que le « 10 » est représenté par un « T » !**
  - Couleur : SHDC pour Pique(Spades), Cœur(Hearts), Carreau(Diamonds), et Trèfle(Clubs)
  - Numéro de la carte : nombre entre 1 et N, avec N la taille du tas initial.
- L'algorithme devra être optimisé afin de traiter plusieurs millions de cartes.

### Exemple

- 39 paquets de 52 cartes sort le **7D2009** correspond au 7 de carreau, qui était à la position 2009 dans le tas initial.
- 18 paquets de 52 cartes sort le **TH849** correspond au 10 de cœur, qui était à la position 849 dans le tas initial.



### Questions

Écrire un programme qui permette de trouver la valeur de la carte restante (rang, couleur et rang dans le paquet d'origine) en fonction du nombre de paquets de 52 cartes.

1. On dispose d'un tas de 52 cartes de jeu représentant 1 paquet.
2. On dispose d'un tas de 2 028 cartes de jeu représentant 39 paquets de 52 cartes
3. On dispose d'un tas de 5 200 000 cartes de jeu, représentant 100 000 paquets de 52 cartes.

## 11. Les bons et les mauvais codeurs (20pts +7/-2)

### Introduction

Un client souhaite une IHM afin de contrôler et administrer un robot qu'il va mettre prochainement sur le marché. Une équipe est en charge des développements. L'équipe n'est pas à plein temps, plusieurs développeurs se relaient, ce qui complique le travail. Bien entendu, dans l'équipe, il y a des bons, et des mauvais codeurs.

Le bon codeur, il voit une page à coder, bon bah, il la code. C'est un bon codeur. Le mauvais codeur, il voit une page à coder, bon bah il la code. Mais c'est un mauvais codeur, rien à voir...



### Règles

- Un bon développeur est capable de créer entièrement  $X_b$  pages en  $Y_b$  jours
- Un mauvais développeur corrompt entièrement  $X_m$  pages en  $Y_m$  jours, ce qui ralentit le bon développeur
- Un codeur développe 8h par jour.
- Tous les codeurs développent en même temps.
- Le but est de calculer en combien de temps les pages seront prêtes.
- Le résultat est attendu au format **00d01h20m30s**
- Le résultat est tronqué à la seconde.
  - Si le résultat est 10,84 secondes, la réponse attendue est **00d00h00m10s**
  - Si les mauvais codeurs empêchent les bons codeurs d'avancer, la réponse attendue est **FAIL**

### Exemples

**Premier exemple :** Il n'y a que deux codeurs pour une page. Le bon codeur fait une page en 1 jour, et le mauvais codeur corrompt une page en 4 jours.

En résumé : 1 page à développer, 1 bon codeur (1j/page), 1 mauvais codeur (4j/page).

La page sera donc prête en 1.333333 jours = 1 jour, 2 heures et 40 minutes.

La réponse attendue est donc **01d02h40m00s**

**Second exemple :** Une équipe de 3 codeurs doivent développer 5 pages. Le premier est un bon codeur, il finit une page en 1 jour. Le second est un mauvais codeur, il corrompt une page en 2 jours. Le troisième est un mauvais codeur, il corrompt une page en 3 jours.

Les 5 pages seront créées en 30 jours.

La réponse est donc **30d00h00m00s**

### Questions

Écrire un programme qui calcule le temps requis aux codeurs pour terminer leur travail

1. 1 page à développer, 1 bon codeur (1j/1page), 1 mauvais codeur (3j/1page)
2. 3 pages à développer, 3 bons codeurs (2j/1page, 3j/1page, 6j/1page), 1 très mauvais codeur (1j/1page)
3. 4 pages à développer, 3 bons codeurs (2j/1page, 3j/1page, 6j/1page), 2 mauvais codeur (5j/2pages, 10 j/3pages)

## 12. Cavalier en cavale (25pts +8/-2)

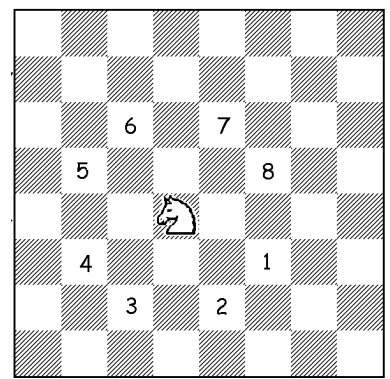
### Introduction

Flash info ! 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 », mais est souvent vu au bar « en bas à gauche » (oui, les propriétaires ont beaucoup réfléchi pour trouver les noms).

### Règles

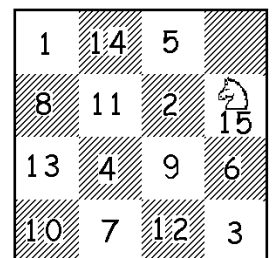
Les instructions sont :

- Le cavalier est sur un échiquier de taille NxM (fourni dans la question)
  - N colonnes, M lignes
- 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 est terminée quand le cavalier n'a plus de bar à foin directement atteignable. Il est dans une impasse, il se fait 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**



### Questions

Écrire un programme qui calcule le temps de cavale sur les plateaux suivants :

1. Plateau 5x5
2. 10x10
3. 1000x1500



## 13. Lecture d'un labyrinthe (25pts +8/-2)

### Introduction

Homer a trouvé le plan de sa cave. Aidez-le à le lire !

### Règles

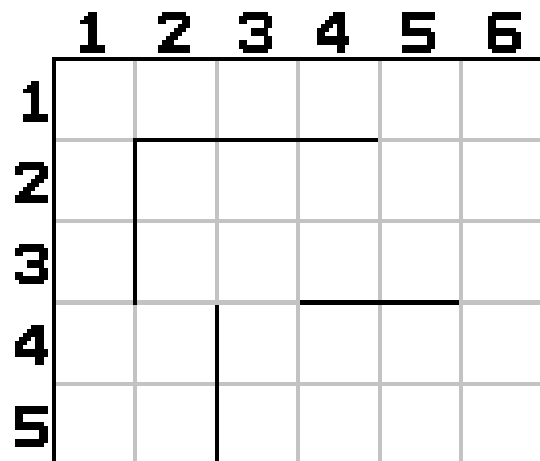
- Pour un labyrinthe de taille NxM (N colonnes, M lignes), la liste des murs est fournie en entrée en précisant les coordonnées d'une cellule, et la direction bloquée par le mur
- Le premier caractère indique quelle direction le mur bloque (Left, Right, Up, Down).
  - L 2 3 => il y a un mur à gauche de la case (2,3) => il y a un mur entre les cases (2,3) et (1,3)
- Le but de l'exercice est de compter le nombre de mur sur chaque case.
- La réponse attendue est le nombre de cases ayant x murs (x vaut 0,1,2,3 ou 4).
- La première coordonnée est X, la seconde Y.
- La case en haut à gauche est la case (1,1)
- Le contour du labyrinthe est un mur



### Exemple

Soit le labyrinthe de taille 6x5 suivant :

R	1	2
L	2	3
D	2	1
U	3	2
D	4	1
L	3	4
R	2	5
D	4	3
U	5	4



Le nombre de cases ayant 1 seul mur est **16**

### Questions

Écrire un programme qui compte le nombre de cases semblables :

1. Sur ce plateau 4x3, donner le nombre de cases ayant 0 murs :
  - U 4 3
  - R 2 3
  - L 2 1
  - L 3 1
  - D 2 2
  - D 3 1
2. Sur le plateau 30x10 du fichier Labyrinth\_q2.txt, donner le nombre de cases ayant 1 mur
3. Sur le plateau 50x150 du fichier Labyrinth\_q3.txt, donner le nombre de cases ayant 2 murs



## 14. Puissance 4 (25pts +8/-2)

### Introduction

Le puissance 4 est un jeu qui se joue à 2 joueurs, et où le but est d'aligner au moins 4 pions de sa couleur, diagonales comprises. Chaque joueur insère à tour de rôle un pion en haut d'une colonne, qui glisse vers la dernière case libre. On vous propose ici d'analyser une partie pour déterminer où sont les jetons gagnants.

### Règles

- Le plateau est de taille LxC (L lignes et C colonnes)
- Les données en entrée sont les numéros de colonnes jouées
  - Une virgule sépare chaque coup
  - Les joueurs s'alternent à chaque coup
  - A chaque coup, on insère un pion dans la colonne annoncée, qui se place sur la dernière ligne encore vide de cette colonne.
- La case en bas à gauche est la case (1,1)
- Le résultat attendu est le jeton du groupe gagnant, au format « **X Y** », qui suit les règles :
  - SW↔NE, la coordonnée demandée est le pion en bas à gauche
  - W↔E, la coordonnée demandée est le pion à gauche
  - NW↔SE, la coordonnée demandée est le pion en haut à gauche
  - N↔S, la coordonnée demandée est le pion en haut
- Le coup gagnant n'est pas obligatoirement le dernier coup joué
- Il n'y a qu'une combinaison gagnante

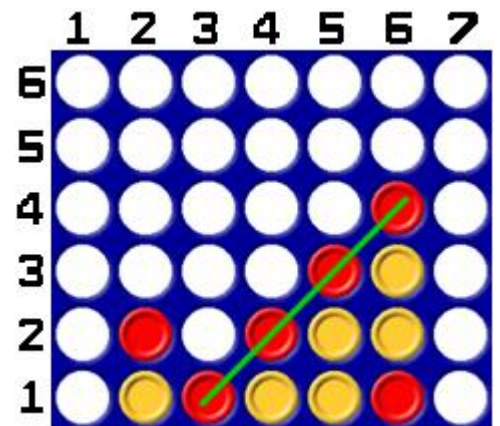
### Exemple

Sur le plateau 6x7 suivant :  
4,3,5,4,5,6,2,2,6,5,6,6

Les pions permettant la victoire sont les pions :  
(3,1),(4,2),(5,3),(6,4)

La configuration des pions est SW↔NE donc c'est le pion en bas à gauche qui est demandé.

La réponse attendue est donc : **3 1**



### Questions

Écrire un programme qui détecte les pions gagnants sur les plateaux suivant :

1. Plateau 6x7 : 6,4,4,4,1,3,6,1,6,6,3,5,1,3,7,4,4,2,6,2,1,4
2. Plateau 6x7 : 6,3,2,3,5,3,4,5,1,6,4,5,3,3,2,3,4,7,1,1,5,2,2,2,6,4,6,4,4,1,6,6,5
3. Plateau 60x70 : Connect4\_q3.txt

## 15. Factures impayées (35pts +12/-2)

### Introduction

Votre père souhaite faire sa comptabilité. N'ayant pas de logiciel lui permettant de le faire de façon automatique, il doit vérifier chaque mois à la main quels sont les mauvais payeurs. Mais heureusement, lors du dernier repas de famille, vous, brillant Sopra-Sterien, lui avez dit que ce ne serait pas compliqué d'automatiser la tâche, à condition de connaître les montants, les dates, ainsi que les numéros de facture... Voulant enfin lui démontrer qu'en fait, vous ne servez pas qu'à résoudre les problèmes d'imprimantes, vous vous êtes même surpris à dire « En même pas une demi heure je te fais ça, p'pa, c'est du gâteau ! ». Vous avez donc 30 minutes pour lui montrer de quoi vous êtes capable ! C'est qui l'patrooonnnn ???

### Règles

- Le fichier d'entrée est une liste **non ordonnée** de crédits et de factures au format A BBB CCC.CC DDDDDDDD
  - A est le caractère « I » comme Invoice (Facture) ou « C » comme Credit (crédit)
  - BBB est le numéro unique de la facture ou du crédit.
  - CCC.CC est le montant de la facture ou du crédit
  - DDDDDDDD est la date de la facture ou du crédit au format AAAAMMJJ
- Le but de l'exercice est de trouver quelles sont les factures non créditées (donc impayées)
- Le crédit a lieu au maximum 3 jours après la facture (bornes incluses)
- Certains crédits ne sont associés à aucune facture. Cela ne doit pas gêner le traitement.
- Il ne peut pas y avoir d'ambiguïté. S'il y a 2 factures du même montant le même jour, il ne peut pas y avoir qu'un seul crédit dans les 3 jours qui suivent. Soit se sont 2 factures payées, soit se sont 2 factures impayées. (voir exemples)
- Un crédit ne doit en aucun cas être associé à plus d'une facture.
- La réponse attendue est la liste des factures impayées, triées par ordre croissant, séparées par une virgule (par exemple : **001,005,123**).

### Exemples :

#### • Simple :

Entrée :

C 001 100.00 20151202  
I 002 110.00 20151205  
C 003 110.00 20151213  
I 004 100.00 20151201  
I 005 110.00 20151210

004 est payée par 001, 005 est payée par 003.

Réponse attendue : **002**

#### • Complexe :

Entrée (ordonnée pour l'exemple) :

I 001 100.00 20151201  
I 002 100.00 20151201  
C 003 100.00 20151201  
C 004 100.00 20151202  
I 005 110.00 20151204  
I 006 110.00 20151205  
I 007 100.00 20151206  
C 008 110.00 20151206  
I 009 100.00 20151206  
I 010 110.00 20151206

001 et 002 sont payées par 003 et 004.

Les factures 005, 006, et 010 précèdent 008.

005 est payée par 008 car c'est la plus vieille.

La réponse attendue est : **006,007,009,010**

Le 4 février 2016

- Cas non autorisé :

Entrée :

I 001 100.00 20151201  
I 002 100.00 20151201  
C 003 100.00 20151202  
I 004 100.00 20151210

Il est ici **impossible** de savoir si c'est la facture 001 ou la 002 qui a été payée par le crédit 003. Ce cas ne se présentera pas dans cet exercice. Par contre, un crédit supplémentaire du type « C 005 100.00 20151204 » résoudrait le problème et rendrait le jeu de donnée valide (même si le crédit 003 n'est pas le même jour !).

### Questions :

- 1) I 001 123.45 20160111  
I 002 123.45 20160102  
I 003 123.45 20160106  
C 004 123.45 20160104  
C 005 123.45 20160114
- 2) C 001 103.42 20151207  
C 002 432.27 20151205  
C 003 845.36 20151202  
I 004 845.36 20151201  
I 005 432.27 20151206  
C 006 845.36 20151201  
I 007 502.91 20151206  
I 008 432.27 20151204  
C 009 432.27 20151206  
I 010 845.36 20151201  
I 011 432.27 20151201



- 3) Fichier Invoices\_q3.txt. Attention, il n'y a pas plus de 5 factures impayées.

## 16. Construction d'un Pick-up Légo automatisé (45pts +15/-2)

### Introduction

Jacky, passionné d'informatique et de nouvelles technologies cherche à créer son propre robot, en utilisant une base Légo, et en y ajoutant les composants intelligents. Il veut donc se procurer une Raspberry Pi, une caméra, des capteurs ultrason, des moteurs, de la mémoire... tout ça tout ça... Le problème, c'est qu'à force d'en parler, le défi est lancé parmi tous ses collègues. Du coup, ils veulent grouper leur achat pour faire des économies ! En plus, Pascal a déjà quelques pièces, de même pour Guy-Noël. Aidez-les à calculer le prix de leur commande.

### Règles

- La liste des pièces dont l'équipe a besoin est fournie en entrée.
- Les pièces sont les suivantes :
  - L = Le pick-up Légo : 100€
  - M = Les moteurs : 60€
  - C = Capteurs, Caméra & Connecteurs : 50€
  - R = La Raspberry Pi 2 : 40€
  - G = Mémoire 16Go classe 10 : 25€
- Le site internet propose les réductions suivantes :
  - MULTIPACK : Pour tout pack L+M+C+R+G acheté, le pack suivant est 15% moins cher que le précédent :
    - 1Pack = 275€
    - 2Packs =  $275+233.75 = 508.75€$
    - 3Packs =  $275+233.75+198.69=707.44€$
    - 4Packs =  $275+233.75+198.69+168.89=876.33€$
  - Lors du calcul du MULTIPACK, chaque prix est arrondi au centime près (0.505 est arrondi à 0.51). Par exemple, le prix calculé du 4<sup>ème</sup> pack est 168.8865, arrondi à 168.89. Le pack suivant se base sur le prix ARRONDI précédent, le prix sera donc  $168.89*0.85=143.5565 \approx 143.56$  (et non  $168.8865*0.85=143.553525 \approx 143.55$ ).
  - PACK DUO+ : Pour 2 pièces achetées, la moins chère des deux est à -50%
    - L+M = 130€
    - L+R = 120€
    - C+C = 75€
    - Etc...
  - Pour chaque article, une seule promotion est applicable. Soit MULTIPACK, soit PACK DUO+, soit aucune.
  - Dans un même panier, les promotions peuvent être mixées.
- Le prix demandé est le prix le plus avantageux possible avec 2 chiffres après la virgule, et avec un « . » comme séparateur. Par exemple : **123.45**
- La question 3 contient un grand jeu de donnée.
  - Y'en a qui ont essayé le brute-force, ils ont eu des problèmes... après, c'est vous qui voyez...
- Cela peut paraître évident, mais en faisant le détail des packs achetés, on doit avoir autant de pièces que dans la liste de départ fournie dans l'énoncé.

Le 4 février 2016

## Exemple

Entrée : LMCRLMM

Solutions possibles :

- $MG + LC + MR + ML + L = 72.5 + 125 + 80 + 130 + 100 = 507.50\text{€}$
- $LMCRG + LM + LM = 275 + 130 + 130 = 535.00\text{€}$
- $LMCRG + LL + MM = 275 + 150 + 90 = 515.00\text{€}$
- $LM + CR + GL + LM + M = 130 + 70 + 112.50 + 130 + 60 = 502.50\text{€}$
- $LM + CR + G + LL + MM = 150 + 130 + 90 + 70 + 25 = 465.00\text{€}$
- Etc...

Ici, le meilleur prix est **465.00**

## Questions

Écrire un programme qui calcule le panier le moins chère avec les pièces suivantes :

1. MLLMLLCLR
2. GRRRLRRGMLMRGGCGMGLMRGCRCCRRLLGRCMCGRCGRLCGLMMLRGLMMLCCGRGR  
MMGRRRCLCR
3. Fichier Pickup\_q3.txt.



## Introduction

## Règles

A 10x10 grid of 100 small, colorful, abstract shapes, each resembling a stylized letter or symbol, arranged in a repeating pattern. The shapes are in various colors including red, blue, green, yellow, and purple, and are set against a white background. The overall effect is a dense, multi-colored mosaic.