



LA COMPÉTITION DE ROBOTIQUE CRC

DÉFIS DE PROGRAMMATION



Un programme de

AEST EAST

ALLIANCE POUR L'ENSEIGNEMENT DE LA SCIENCE ET DE LA TECHNOLOGIE
EDUCATIONAL ALLIANCE FOR SCIENCE AND TECHNOLOGY

Table des matières

CONTEXTE.....	2
Problème #1 - La Division Entière (1 point)	3
Problème #2 - Le modèle du vaisseau (2 points)	5
Problème #3 - Simuler des interrupteurs (3 points)	7
Problème #4 - Mesure de la température (4 points)	8
Problème #5 - Températures habitables (5 points).....	9
Problème #6 - Les Codes-Barres (6 points)	10
Problème #7 - Équilibrer les propulseurs (7 points)	11
Problème #8 - Calcul de distance (8 points)	13
Problème #9 - Sélection de manettes (9 points).....	15
Problème #10 - Partir le réacteur (10 points)	17
Problème #11 - Contrôle de la Consommation (11 points)	19
Problème #12 - Arrosage des plantes (12 points).....	21
Problème #13 - L'Encyclopédie (13 points)	23
Problème #14 - Encoder un message (14 points).....	25
Problème #15 - Calculateur de Nombre Complexes (15 points)	28

CONTEXTE

Nous sommes en 2087. Après l'échec de la communauté internationale de réduire la production de gaz à effet de serre, la Terre a passé un point de non-retour et devient de plus en plus inhabitable à chaque minute qui passe. Les marées montent, les continents rétrécissent et les pandémies se multiplient. Cependant, il nous reste un espoir: vous. Vous avez découvert une exoplanète potentiellement habitable dans un système planétaire suffisamment proche.

En chemin vers cette nouvelle planète qui pourrait tous nous sauver, vous avez percuté un astéroïde massif, ce qui a causé votre écrasement sur la planète la plus proche, vous coûtant votre seul ingénieur-mécanicien. La planète est composée principalement d'une roche étrange inconnue à ce jour. Ainsi, pendant que votre robot d'expédition récolte des échantillons de cette pierre qui pourrait faire avancer la science, vous devrez réparer les différentes parties du vaisseau qui ont été endommagées.

Votre tâche sera donc de réparer ces composantes du vaisseau le plus rapidement possible afin de repartir le plus tôt possible. Si vous faites tout correctement, vous devriez pouvoir arriver en février l'année prochaine. Dépêchez-vous, astronautes, vous pouvez encore sauver la race humaine...

Problème #1 - La Division Entière (1 point)

Parfois, quand on divise des entiers, on obtient une réponse décimale et dans certains cas, cette réponse peut ne pas nous intéresser, comme si on séparait un panier d'oranges également en plusieurs paniers plus petits, mais qu'on ne voulait pas séparer les oranges elles-mêmes. Il est donc possible que, non seulement le résultat entier nous intéresse, mais aussi ce qu'il reste à la fin (ou ce qu'on ne peut plus séparer).

Au lieu de simplement effectuer la division, il faudra donc soustraire le diviseur du dividende jusqu'à ce qu'on ne puisse plus le faire. Le nombre de soustractions effectuées devient le résultat de la division et le nombre qui reste après qu'on ne puisse plus soustraire devient le reste. Il suffit de visualiser que le programme effectue ce que nous effectuerions normalement lors d'une division à crochet. Nous aurons ainsi le résultat entier et le reste de la division, qui est lui-même très utile en programmation.

Spécification d'entrée

La première ligne correspond au nombre de tests à effectuer. Les lignes suivantes contiennent chacune deux nombres séparés d'un espace. Le premier est le dividende, ou le nombre à diviser, et le deuxième est le diviseur, ou le nombre par lequel on divise l'autre.

Spécification de sortie

Pour chaque division, il faudra donner la réponse sous la forme "a reste b" où a est le résultat et b est le reste. S'il n'y a pas de reste ($b = 0$), on donne juste le résultat.

Exemple d'entrée

```
5
56 18
34 7
89 128
581397 869
1159029 9423
```

Exemple de sortie

```
3 reste 2
4 reste 6
0 reste 89
669 reste 36
123
```

Explication de la première sortie

On soustrait 18 de 56 une fois, nous laissant 38. On peut ensuite le soustraire une deuxième fois, ce qui nous laisse 20. On refait pareil pour avoir 2. Maintenant 18 ne peut plus être soustrait, donc 2 devient le reste. 18 a été soustrait 3 fois dans le processus, ce qui donne 3 reste 2. On peut en effet vérifier avec $18 \times 3 + 2 = 56$.

Problème #2 - Le modèle du vaisseau (2 points)

Votre tâche sera ici de créer le programme qui vous permet de visualiser une version miniature de votre vaisseau sur un écran dans la salle de pilotage. Cette petite version de votre vaisseau vous sert surtout à voir rapidement et facilement s'il y a un problème ailleurs dans le vaisseau en désignant la zone et le problème sur le modèle.

Heureusement, il ne vous faudra que créer "le squelette" et les données qui ont été sauvées du programme s'occuperont du reste.

Ainsi, le "squelette" est une seule forme fermée composée d'un rectangle avec un triangle sur le dessus ou, si on se l'imaginait en trois dimensions, un cylindre doublé d'un cône sur le dessus.

Spécification d'entrée

La première ligne indique le nombre de tests effectués pour tester votre programme. Toutes les lignes suivantes contiennent deux chiffres entre 1 et 10 séparés par un espace, soit respectivement la hauteur du rectangle, puis la hauteur du triangle.

Note: Il est important de remarquer que, parce que la sortie dans la console d'un problème est habituellement monospace (toutes les lettres et tous les symboles prennent exactement le même espace), la hauteur du triangle correspond donc aussi à la moitié de sa largeur et donc la moitié de la largeur du rectangle (moins les côtés).

Spécification de sortie

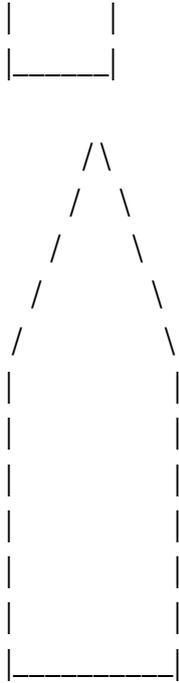
La sortie sera, pour chaque série de dimensions (chaque ligne en entrée), un "dessin" formé de "_" et de "|" pour le rectangle, et de "/" et de "\" pour le triangle.

Exemple d'entrée

```
2
5 3
7 5
```

Exemple de sortie

```
  /\
 /  \
/    \
|      |
|      |
|      |
```



Explications de la première sortie

5 est spécifié pour la hauteur du rectangle, ce qui fait que les côtés du rectangle contiennent 5 “|”. Ensuite, 3 est spécifié pour la hauteur du triangle, ce qui fait que la largeur est de 6 excluant les murs (il faut 3 caractères pour atteindre la hauteur et ainsi 3 pour en redescendre). Nous avons donc 3 “/” ou “\” qui constituent chaque côté du triangle, puis la largeur du rectangle en bas doit correspondre : il y a donc 6 “_”.

Problème #3 - Simuler des interrupteurs (3 points)

Dans n'importe quel contexte, beaucoup de programmes doivent simuler le fonctionnement d'un interrupteur ou d'une "switch" pour activer ou désactiver certaines fonctionnalités. Un tel interrupteur, s'il était physique, aurait deux positions pour le levier ou le bouton qui l'active : une position qui veut dire que l'interrupteur est allumé et une position qui veut dire que l'interrupteur est éteint.

Maintenant, une manière très efficace de simuler ceci en programmation est d'utiliser un booléen (variable qui peut être True ou False). Ainsi, quand on "bouge" le levier, s'il était False, il devient True, et s'il était True, il devient False.

Il vous faudra donc simuler 3 de ces switches simultanément, de sorte qu'on puisse activer ou désactiver n'importe laquelle des 3 dans n'importe quel ordre.

Spécification d'entrée

La première ligne correspond au nombre de tests à effectuer. Chaque ligne suivante correspond à une liste de commandes à effectuer toutes séparées par des espaces. Les commandes sont S1, S2 et S3, et correspondent à leur levier respectif. **Par défaut, les switches sont désactivés au début de chaque test.**

Spécification de sortie

Pour chaque série de commandes, vous devrez envoyer le résultat de chacune des switches une fois que toutes les commandes auront été effectuées. Le résultat sera portrayé par True ou False et les switches devront être dans l'ordre.

Exemple d'entrée

```
3
S1 S3
S2 S1 S2 S3 S3
S3 S1 S1 S2 S3 S2 S3 S1 S2 S1
```

Exemple de sortie

```
True False True
True False False
False True True
```

Explication de la première sortie

Tous les switches sont à False au début, puis on active la première et la troisième (une seule fois donc on ne la rabaisse pas), nous laissant finalement avec True False True.

Problème #4 - Mesure de la température (4 points)

Pour opérer efficacement le vaisseau, il faut mesurer la température. Cependant la compagnie ayant créé le thermomètre est américaine et donne la mesure en fahrenheit. L'ordinateur de bord doit faire la conversion vers les kelvins, une unité beaucoup plus scientifique.

Pour convertir une mesure en Fahrenheit en Celsius, la formule est : $C = 5 (F - 32) / 9$

Pour convertir une mesure en Celsius en Kelvin, la formule est : $K = C + 273.15$

Votre programme doit prendre une mesure en Fahrenheit et la convertir en Kelvin.

Spécification d'entrée

La première ligne contient un entier $1 \leq N \leq 100\,000$ qui indique le nombre de mesures prises par le capteur. Les N prochaines lignes contiennent un réel P qui représente la mesure en Fahrenheit.

Spécification de sortie

Pour chaque ligne, la sortie est un nombre réel avec une précision d'un chiffre décimal correspondant à la température en Kelvin.

Exemple d'entrée

3
1.0
-3.7
100.0

Exemple de sortie

255.9
253.3
310.9

Explication de la première sortie

Par la première équation, $1.0^\circ\text{F} = -17.2^\circ\text{C}$ et, par la deuxième équation, $-17.2^\circ\text{C} = 255.8^\circ\text{K}$.

Problème #5 - Températures habitables (5 points)

Pour trouver une planète habitable, le détecteur d'exoplanètes prend les températures de planètes dans l'univers à différents endroits et moments sur celles-ci pour obtenir six mesures différentes. Pour s'assurer que les températures sont correctes sur cette planète, votre programme devra donner la température moyenne avec 1 décimale de précision. Si la température n'est pas convenable pour un humain, il faut retourner le message "danger". Pour que la température de la planète soit valide, il faut que la moyenne soit entre 10 et 30 (inclusivement) degrés Celsius, et que le plus grand écart entre deux températures mesurées ne dépasse pas 60 degrés.

Il faudra trouver la température moyenne et s'assurer qu'elle est acceptable pour un humain.

Spécification d'entrée

La première ligne indique un entier n entre 1 et 10 000, correspond au nombre de tests effectués. Les n prochaines lignes contiennent 6 réels, soit les mesures de température sur la planète.

Spécification de sortie

La sortie attendue est un réel arrondi à 1 décimale si la moyenne et le plus grand écart respectent les normes. Sinon, la valeur attendue est "danger".

Exemple d'entrée

```
3
4.3 2.3 51.3 34.5 29.0 12.4
-12.5 2.9 17.6 -16.3 7.7 20.5
-24.0 42.4 26.5 29.9 17.5 17.6
```

Exemple de sortie

```
22.3
danger
danger
```

Explication de la première sortie

22.3 correspond à la moyenne de la température, car celle-ci se situe dans les standards acceptables pour la température moyenne, et l'écart maximal de température est de 49 degrés Celsius, ce qui est aussi correct. On respecte les deux conditions, donc on donne la température moyenne.

Problème #6 - Les Codes-Barres (6 points)

Dans le vaisseau, tous les artefacts et échantillons trouvés en mission doivent être classifiés à l'aide d'un code-barres qu'on pourra scanner pour accéder à l'information rapidement dans la base de données. Un numéro précis est attribué à chaque échantillon à la collecte. Ce numéro est donné selon notre système décimal classique. Il vous faudra d'abord le convertir en nombre binaire 16 bits, puis changer les 0 et les 1 pour, respectivement, des espaces et des barres.

Spécification d'entrée

La première ligne est un entier représentant le nombre de tests à effectuer. Chaque ligne qui suit correspond à un test et est constituée d'un nombre entier positif n entre 1 et 65535.

Spécification de sortie

Votre code devra donner le code-barres correspondant au numéro de chaque test. Des # encadreront les codes-barres pour clarifier le début et la fin de l'étiquette.

Exemple d'entrée

```
3
17238
9819
39385
```

Exemple de sortie

```
# | ||| |#
# | || | || |#
#| || ||| |#
```

Explication de la première sortie

17238 donne 0100001101010110 en binaire 16 bits et, en remplaçant les 1 par des barres et les 0 par des espaces, on obtient la première ligne ci-dessus.

Problème #7 - Équilibrer les propulseurs (7 points)

Le système de propulsion du vaisseau est composé de deux réacteurs perpendiculaires. Le réacteur principal A s'occupe de la poussée vers l'avant, alors que le réacteur B s'occupe de la poussée latérale. Le système de l'ordinateur de bord est un prototype qui n'a pas été mis à jour avec le système de la fusée. Par conséquent, il donne les coordonnées de navigations en contrôle "tank", comme si les 2 fusées propulsaient dans la même direction.

Vous devez convertir les commandes de navigation d'une entrée au format (Gauche, Droite) vers le format (Avant, Latéral). Par convention, le sens positif du réacteur latéral fera tourner le vaisseau vers la droite. On cherche donc à convertir des commandes du format **(a, a+b)**, où b symbolise simplement une différence de poussée entre les deux réacteurs, au format **(2*[réacteur le plus faible], b)**. En regardant plus attentivement, on voit en effet que les poussées égales font avancer le vaisseau alors que le surplus d'un côté fait tourner le vaisseau, ce qui est converti en coordonnée latérale.

Spécification d'entrée

La première ligne contient un entier $1 \leq N \leq 10\,000$ qui indique le nombre de tests à effectuer. Les prochaines N lignes contiennent un réel 'G', la force de poussée de gauche, et un réel 'D', la force de poussée de droite. La précision d'entrée est au dixième près.

Spécification de sortie

Pour chaque entrée, la sortie attendue est un réel 'A' qui décrit la poussée avant et un réel 'L' qui décrit la poussée latérale. La précision attendue est aussi au dixième près.

Exemple d'entrée

```
5
12.2 13.4
23.2 0.0
3.7 3.7
-1.0 -1.2
2.1 2.2
```

Exemple de sortie

```
24.4 1.2
0.0 -23.2
7.4 0.0
-2.0 -0.2
4.2 0.1
```

Explication de la première sortie

Nous avons 1.2 de puissance en plus dans le réacteur de droite, ce qui nous donne une puissance de +1.2 pour le réacteur latéral. Ensuite, les 12.2 restants de chaque côté s'équilibrent en direction pour faire avancer le vaisseau de 24.4 vers l'avant.

Problème #8 - Calcul de distance (8 points)

Vivant dans un monde en trois dimensions, il nous faut trois coordonnées pour décrire une position unique, peu importe le système de coordonnées utilisé. Il y a des systèmes de coordonnées plus compliqués qui sont mieux adaptés pour l'espace et l'astronomie, mais nous utiliserons ici les coordonnées cartésiennes étant le système le plus simple et le plus connu.

Ainsi, tout point dans tout espace en trois dimensions peut être exprimé par des coordonnées (x,y,z) . Cependant, le vaisseau ne parcourra pas le chemin jusqu'à un point de coordonnées en parcourant les axes un à un car ce serait beaucoup plus lent. Ainsi, on cherche ici à calculer la distance que le vaisseau devra parcourir en ligne droite entre deux positions.

En sachant que les axes x , y et z sont perpendiculaires et en faisant un peu de trigonométrie, on peut voir que la distance totale peut être calculée ainsi :

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

où Δx est la distance entre les deux coordonnées en x ($x_1 - x_2$). C'est la même chose pour les y et les z .

Finalement, dépendamment si on est très loin ou extrêmement proche de notre cible, on aimerait potentiellement avoir plusieurs unités de distance différentes à notre disposition.

Les unités utilisées seront donc le kilomètre (km) pour les petites distances, l'unité astronomique (UA), soit la distance Terre-Soleil, pour les distances moyennes, et l'année-lumière (AL), soit la distance que la lumière parcourt en une année, pour les distances les plus longues. Il existe des unités encore beaucoup plus grandes mais nous n'en traiterons pas ici.

Les taux de conversion que vous utiliserez seront les suivants :

$$\begin{aligned}1 \text{ UA} &\approx 150\,000\,000 \text{ km} \\1 \text{ AL} &\approx 9\,461\,000\,000\,000 \text{ km} \\1 \text{ AL} &\approx 63\,073 \text{ UA}\end{aligned}$$

Spécification d'entrée

La première ligne contient le nombre de tests à effectuer. Les lignes suivantes contiennent trois données principales. D'abord, la première donnée correspondra à l'unité de mesure dans laquelle les coordonnées seront données. Ensuite, la deuxième donnée sera la position du vaisseau dans le système de coordonnées cartésiennes. Finalement, la dernière

donnée sera la position du point où on veut se rendre et avec lequel on devra calculer la distance.

Spécification de sortie

En sortie, on exigera la distance entre les deux points **exprimés dans l'unité la plus appropriée et arrondi au millième près**. Des différences de ± 0.003 unités sont acceptées.

Distance d	Unité exigée
$d < 0.1 \text{ UA}$	Kilomètre (km)
$0.1 \text{ UA} \leq d < 0.1 \text{ AL}$	Unité Astronomique (UA)
$0.1 \text{ AL} \leq d$	Année-lumière (AL)

Exemple d'entrée

5

km (27486 53294 2435) (57293 189070 289065)

UA (-0.37 1.27 8.92) (0.77 -2.82 5.61)

km (-302 9047 -65) (-257988303 287302 807944832)

AL (2.5608 3.8729 -0.0192) (2.5712 3.8477 -0.0185)

AL (234 -845 99) (58 -1133 -202)

Exemple de sortie

318559.785 km

5.384 UA

5.654 UA

1720.044 UA

452.240 AL

Explication de la première sortie

On met le tout dans la formule de distance, ce qui nous donne 318559.785 km. Étant donné que 0.1 UA correspond à environ 1 500 000 km, nous sommes en dessous et restons en km pour la valeur sortie.

Problème #9 - Sélection de manettes (9 points)

Pour activer la plus grande vitesse de votre vaisseau, vous devez atteindre la pleine puissance des réacteurs en activant toutes les manettes de la salle des moteurs. Comme les commandes sont disposées en rond, vous les activez en partant de la deuxième manette, car la première est trop difficile à activer. Vous les activez ensuite dans l'ordre croissant des manettes. Par contre, quand vous actionnez une manette, vous ne pouvez pas tout de suite activer la suivante, car vous recevrez une décharge électrique. Vous devez donc toujours éviter la manette suivante à celle que vous venez d'activer et ce, même si les nombres pour marquer les manettes ne se suivent pas. Comme la pièce est circulaire, après un premier passage vous continuez jusqu'à revenir aux premiers chiffres et repartez du début en cercle selon la même logique de toujours sauter une manette. Le compte des manettes va de 1 jusqu'à la valeur donnée inclusivement. Par contre, on commence en activant la deuxième manette d'abord.

Vous cherchez donc à savoir quelle sera la dernière manette à être activée dépendamment du nombre de manettes dans la salle des moteurs.

Spécification d'entrée

La première ligne indique le nombre de cas de tests à faire. Les lignes suivantes contiendront un entier représentant le nombre de manettes dans la salle des moteurs.

Spécification de sortie

La sortie sera un entier correspondant à la dernière manette à activer sans s'électrocuter (en suivant la règle d'une manette sur deux).

Exemple d'entrée

3
13
9
8

Exemple de sortie

11
3
1

Explication de la première sortie

On fait d'abord un tour de table, enlevant tous les nombres pairs, ce qui laisse les manettes [1, 3, 5, 7, 9, 11, 13]. Après être passé sur 12, on esquive 13 et on abaisse la première manette.

De la même manière, on abaisse 5, 9, et 13. On doit maintenant recommencer un troisième tour du début avec les manettes [3, 7, 11]. On évite 3 pour abaisser 7. Finalement, on saute 11 et on baisse 3, nous laissant avec 11 comme dernière manette à activer.

Problème #10 - Partir le réacteur (10 points)

Vous êtes à bord de votre vaisseau spatial pour un très long voyage, ce qui requiert que votre vaisseau ait sa propre source d'énergie. Le vaisseau possède donc un assez gros réacteur (de type inconnu, mais cela vous importe peu) qui est capable de fournir énormément d'énergie.

Cependant, ce réacteur s'est malencontreusement éteint lors du crash. Parce qu'il est potentiellement dangereux, il y a un code de sécurité à 20 chiffres pour le repartir. Vous voyez aussi qu'il y a deux afficheurs numériques sur le réacteur : un, à gauche, qui peut afficher jusqu'à 4 chiffres et un autre, à droite, qui peut n'en afficher que deux.

Ayant assisté à une grande partie de la confection de ce réacteur, vous savez qu'il s'agit de la suite de Syracuse. L'indicateur de gauche indique c'est la suite de Syracuse de quel nombre, et celui de droite indique d'où partir la suite dans le code de sécurité. Il est important de concevoir un programme qui peut calculer le tout, car les deux indicateurs, et donc le code de sécurité, changent toutes les 20 secondes, ce qui laisse très peu de temps pour un humain de calculer le tout.

La suite de Syracuse est donc une vieille suite mathématique qui prend n'importe quel entier et finit toujours par le ramener à 1. On procède donc comme suit pour chaque instance (nombre) de la suite :

- Si l'entier est pair, on le divise par deux
- Si l'entier est impair, on le multiplie par trois et on ajoute 1 au tout

Il faudra donc faire un programme qui calcule la suite de Syracuse pour un certain nombre n et l'affiche (sans espace entre chaque instance) à partir du nombre affiché à droite (on prendra le nombre n comme étant l'instance #1). Il est aussi important de noter que la suite de Syracuse ne prend pas fin lorsque le chiffre 1 est atteint mais continue jusqu'à l'infini avec la période 1 4 2.

Spécification d'entrée

La première ligne correspond au nombre de tests effectués. Les lignes suivantes correspondent aux deux paramètres affichés sur les indicateurs, dans l'ordre dans lequel ils ont été énoncés plus haut.

Spécification de sortie

Votre programme devra sortir une série de vingt chiffres qui servira de code secret pour activer le réacteur.

Exemple d'entrée

4

0014 05

2587 22

8191 85

9999 37

Exemple de sortie

34175226134020105168

20741037311215567783

20230101153034615173

31995847914387192158

Explication de la première sortie

On demande la suite de Syracuse du nombre 14, qui va comme suit :

14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2 ...

On veut construire un code de sécurité de 20 caractères à partir du 5^e chiffre de la suite, soit 34. On a donc le code 34175226134020105168 où on s'arrête, car on a atteint le 20^e chiffre du code.

Problème #11 - Contrôle de la Consommation (11 points)

Maintenant, vous devez vous assurer que votre vaisseau ne consomme pas trop d'électricité. Pour ce faire, un tableau contenant le nombre d'heures actives par jour de chaque salle vous sera donné. On vous donnera aussi un tableau contenant la consommation horaire de chaque pièce.

Ainsi, vous devrez d'abord déterminer la consommation **totale** de chaque pièce au cours de la période donnée, puis vous devrez déterminer la consommation totale de tout le vaisseau. La consommation horaire est en Watts (W) ce qui, avec le nombre d'heures, devrait vous donner des watts-heure (Wh).

Spécification d'entrée

La première ligne contient les dimensions du tableau des heures actives (max 50 x 50), de sorte que le premier chiffre est le nombre de rangées (soit le nombre de jours) et le deuxième chiffre correspond au nombre de colonnes (soit le nombre de pièces). Ensuite, ce tableau vous est donné directement en dessous. Finalement, le deuxième tableau, soit le tableau des consommations horaires de chaque pièce, vous est donné à la suite du premier tableau. Il est important de noter que le deuxième tableau comporte autant de colonnes que le premier où le tout n'aurait pas de sens.

Spécification de sortie

Il vous faudra fournir un tableau contenant la consommation totale de chaque pièce pendant la période donnée. Après, en dessous de ce dernier, vous devrez fournir un chiffre symbolisant la consommation totale du vaisseau en Wh (donc la somme du tableau que votre code vient de donner). Toutes vos sorties doivent être arrondies au Wh près. Des différences de quelques Wh sont acceptées.

Exemple d'entrée

14 17

4.23 22.4 11.2 13.0 19.0 7.25 20.8 11.1 4.99 5.55 10.9 8.09 16.7 3.10 11.4 19.9 5.41
2.14 1.17 15.4 13.0 8.92 2.20 17.8 20.2 17.9 7.88 4.02 20.0 4.21 5.98 1.52 1.85 18.8
13.7 2.92 0.96 21.5 13.4 19.3 23.2 11.2 18.1 21.7 21.5 22.9 20.5 20.0 12.2 23.3 16.5
4.59 13.2 1.46 18.9 20.7 5.68 7.85 22.5 16.9 2.84 16.1 19.0 21.6 0.61 18.1 0.88 11.7
14.9 23.7 6.51 0.70 23.1 3.57 21.6 11.4 2.35 20.8 12.3 2.37 6.29 13.3 3.15 22.0 22.3
8.03 3.77 7.81 11.3 23.1 8.57 20.9 14.8 19.6 6.08 18.3 0.29 12.3 10.8 22.6 22.5 11.0
1.64 18.1 0.43 18.8 5.86 10.2 19.1 21.3 1.82 14.1 10.7 4.31 13.8 18.9 18.5 17.4 18.4
18.0 1.75 15.7 18.9 3.55 20.3 9.66 16.8 15.0 3.86 8.76 19.1 17.5 4.77 11.1 9.67 17.2
20.0 4.46 16.7 10.2 6.09 18.5 14.6 14.7 9.14 16.8 21.6 9.99 20.5 19.5 7.78 16.5 5.58
9.44 7.63 14.9 17.8 1.86 18.9 13.9 12.3 11.5 5.48 17.6 2.77 8.03 24.0 12.9 23.8 8.43

5.70 2.29 8.76 14.6 18.2 3.74 7.43 15.9 7.65 16.1 5.46 20.9 17.3 13.1 5.00 0.33 4.06
2.34 16.9 17.7 12.8 8.25 12.1 9.15 3.92 21.5 0.33 11.8 1.49 16.4 2.04 12.2 6.20 0.45
19.1 15.3 14.8 4.46 18.4 0.04 23.7 6.59 22.3 18.9 12.0 4.18 9.47 14.9 18.9 19.0 22.4
5.52 6.96 3.15 0.89 6.59 3.07 15.7 1.50 6.17 10.7 19.8 7.26 15.2 19.2 8.32 1.93 11.0
10 25 60 60 100 100 100 100 100 350 550 850 1200 2000 2000 2300 3000

Exemple de sortie

1293 3514 8129 10611 17702 13342 22539 18421 17492 52892 104962 121252 239760 340400
327340 426098 519690
2245437 Wh

Explication de la sortie

On additionne toutes les heures de consommation de la première pièce, soit la première colonne, on multiplie par sa consommation horaire et on arrondit au Wh près, ce qui donne 1293 Wh. On fait de même pour toutes les colonnes, puis on les ajoute ensemble pour obtenir la consommation totale du vaisseau en Wh, soit 2245437 lorsqu'arrondi.

Problème #12 - Arrosage des plantes (12 points)

Pour nourrir l'équipage, une serre est présente sur le vaisseau. Il faut arroser chaque plante selon une quantité déterminée par l'ordinateur de bord. Cependant, seuls deux pots sont disponibles et la quantité d'eau à verser n'est pas une des quantités de ces deux pots. Il est également impossible de mesurer une quantité partielle (impossible d'ajouter 1 litre d'eau dans un pot de 2 litres).

À chaque étape, vous pouvez effectuer les commandes suivantes :

- Remplir le seau 1 (FILL1)
- Remplir le seau 2 (FILL2)
- Transvider le contenu du seau 1 dans le seau 2 (POUR1)
- Transvider le contenu du seau 2 dans le seau 1 (POUR2)
- Vider le seau 1 (DISCARD1)
- Vider le seau 2 (DISCARD2)

Spécification d'entrée

La première ligne indique un entier $1 \leq N \leq 10\,000$, le nombre de tests décrits. Les prochaines N lignes contiennent un entier 'Q1', la taille en litres du premier seau, 'Q2', la taille en litres du second seau, et un entier 'T', la quantité d'eau désirée dans un des contenants. Les données sont telles que $1 \leq 'Q1' < 'Q2'$ et $2 \leq 'T' \leq 'Q2'$.

Spécification de sortie

Chaque ligne correspond à la séquence optimale d'opérations pour arroser la plante. Pour chaque opération, indiquer l'action selon le nom spécifié dans l'énoncé du problème. À la fin de la solution, un des deux pots doit contenir la quantité d'eau demandée.

Exemple d'entrée

```
3
3 5 4
7 23 5
39 80 2
```

Exemple de sortie

```
FILL2 POUR2 DISCARD1 POUR2 FILL2 POUR2
FILL1 POUR1 FILL1 POUR1 FILL1 POUR1 FILL1 POUR1 DISCARD2
FILL2 POUR2 DISCARD1 POUR2
```

Explication de la première sortie

Les 2 seaux ont initialement 0L et 0L. On remplit le second seau, ce qui donne 0L et 5L. On vide le second seau dans le premier, ce qui donne 3L et 2L. On vide le premier seau, pour avoir 0L et 2L. On vide le second seau dans le premier, ce qui donne 2L et 0L. On remplit le second seau, ce qui donne 2L et 5L. On verse du second seau dans le premier, ce qui donne 3L et 4L. Le second seau contient 4L, donc la solution est valide.

Problème #13 - L'Encyclopédie (13 points)

Durant votre séjour sur la planète, vous avez un robot qui sera à vos côtés pour récolter des échantillons de matière sur la planète. On cherche donc ici, à l'aide d'une encyclopédie interne, à identifier de quels matériaux sont composés certains échantillons et en quelles proportions.

Une encyclopédie contenant de l'information importante sur plusieurs matériaux vous sera donc donnée et vous devrez, à partir des informations de votre échantillon, donner sa composition. **Chaque échantillon ne sera composé que de deux matériaux différents.**

Spécification d'entrée

La première ligne contient un entier n qui correspond au nombre d'entrées (et donc de matériaux) dans le dictionnaire. Les lignes suivantes contiennent chacune une entrée du dictionnaire de la forme [Nom - Volume - Masse pour le volume indiqué - Couleur], soit les propriétés du matériau. **Deux matériaux différents n'auront pas la même couleur.**

La ligne après le dictionnaire correspond au nombre m de tests à effectuer (ou au nombre d'échantillons à identifier). Les lignes suivantes contiennent chacune un test de la forme [Couleurs - Volume - Masse], soit les propriétés de l'échantillon.

Les masses sont en g et les volumes sont en cm^3 dans le problème.

Spécification de sortie

Pour chaque échantillon on s'attend à un pourcentage arrondi au centième près de chacun des deux matériaux présents de la forme [X% matériau1 et Y% matériau2]. La composition est représentée volumiquement, c'est-à-dire qu'un échantillon est fait à 5% d'un matériau si ce matériau prend 5% du volume de l'échantillon. Des différences de quelques centièmes de pourcentage sont acceptées. **On donne le matériau le plus présent dans l'échantillon d'abord (plus haut pourcentage).**

Exemple d'entrée

4

Oxyde de Fer - 300 - 1572 - Rouge

Quartz - 171 - 453.32 - Rose

Or - 89 - 1717.7 - Jaune

Argent - 237 - 2486.13 - Gris

3

Rouge/Jaune - 2689.22 - 17267.99

Rose/Gris - 394.55 - 3677.58

Jaune/Gris - 578.24 - 8894.87

Exemple de sortie

91.60% Oxyde de Fer et 8.40% Or

85.09% Argent et 14.91% Quartz

55.54% Or et 44.46% Argent

Explication de la première sortie

Comme on retrouve du jaune et du rouge dans l'échantillon, on peut en déduire qu'il est composé d'oxyde de fer et d'or. On peut ensuite trouver les proportions à l'aide du volume et de la masse de l'échantillon, et des informations du dictionnaire interne. Il ne reste plus qu'à arrondir le résultat et mettre l'oxyde de fer devant, car il est plus représenté dans l'échantillon que l'or.

Problème #14 - Encoder un message (14 points)

Étant donné que les ondes électromagnétiques comme la lumière se transmettent aussi bien dans l'espace que dans l'air, il vous serait possible d'envoyer des messages (sur des ondes radio, par exemple) à la Terre.

Maintenant, imaginez qu'il n'y a aucune planète entre vous et votre destinataire (la Terre) et ainsi rien ne pourra bloquer le passage des ondes. Vous pouvez donc transmettre des messages à la Terre et votre vaisseau en est capable. Le problème, c'est qu'il faudrait encoder les messages pour éviter qu'ils soient compris de tout le monde qui les perçoit.

Dans ce problème, vous réparerez donc l'encodeur de message du vaisseau. Il faudra aussi que la clé avec laquelle vous encodez votre message soit comprise de son récepteur. Vous utiliserez donc, comme convenu au décollage du vaisseau, les dates d'envoi et de réception du message (qui seront calculées pour vous) pour modifier la valeur des lettres. Voici comment vous calculerez la nouvelle lettre:

$$\#L = (D_{\text{envoi}}^*[i] \times \#L + D_{\text{réception}}[i]) \bmod 26$$

où $\#L$ est le numéro associé à la lettre (A=>1, B=>2 ... Z=>26),

$D_{\text{envoi}}^*[i]$ est le i -ème chiffre de la date d'envoi corrigée (voir plus bas),

$D_{\text{réception}}[i]$ est le i -ème chiffre de la date de réception (voir plus bas)

et $\bmod 26$ marque le reste d'une division entière par 26 (car si le résultat dépasse 26, on n'est plus dans l'alphabet).

Pour les dates, le format utilisé sera AAAA/MM/JJ. Ainsi, quand on parcourt les lettres d'un message (on ne compte pas les espaces ni les caractères spéciaux comme le .), on utilise le premier chiffre de l'année des deux dates pour encoder la première lettre, le deuxième chiffre de la date pour encoder la deuxième et ainsi de suite jusqu'à la neuvième lettre, où on recommence le cycle avec le premier chiffre de l'année. (On considérera donc que $1000 \leq \text{année} \leq 9999$)

Cependant, il y a un problème majeur avec cette méthode. Si le chiffre de la date d'envoi utilisé n'est **pas** premier avec 26, au moment du décodage, la nouvelle lettre peut mener à plusieurs réponses. **Ainsi, le chiffre utilisé pour la date d'envoi devra toujours être premier avec 26 et ne pourra donc être que 1, 3, 5, 7 ou 9. Si certains des chiffres donnés dans la date d'envoi sont incompatibles, ils devront être corrigés à l'entier supérieur.** (ainsi, un 2 dans la date d'envoi donne 3)

Spécification d'entrée

La première ligne vous fournira donc deux dates suivant le format AAAA/MM/JJ. Il vous faudra faire abstraction des espaces. La date de gauche sera la date d'envoi et celle de

droite, la date de réception du message sur la Terre (oui, calculé d'avance). Ensuite, à partir de la deuxième ligne, vous aurez le message qu'il vous faudra encoder. **Il vous faudra conserver tous les caractères spéciaux du message tout en les ignorant pour l'encodage. Le message vous sera donné sans accents sur les lettres pour simplifier. Finalement, les seuls caractères que vous devrez ignorer et renvoyer dans le message encodé sont [, .] et l'espace.**

Spécification de sortie

Il vous faudra fournir le message encodé en suivant exactement la même structure que le message que vous avez reçu tout en ayant changé toutes les lettres suivant la formule d'encodage. **Les majuscules doivent être conservées.**

Exemple d'entrée

3185/11/25 3188/05/18

Vaisseau Invicta a la base terrestre. Nous avons trouve une exoplanete potentiellement habitable assez proche de notre location. Cependant, nous avons ete frappes par un asteroide, nous forcant a nous ecraser sur une planete environnante. Il semble aussi qu'il puisse y avoir presence de vie sur la planete sur laquelle nous nous trouvons. Je, attitre capitaine du vaisseau Invicta, m,occupe donc de reparer le vaisseau pendant que l,equipe de chercheurs recolte des echantillons pour tenter de confirmer ou d,infirmier cette possibilite. Informations supplementaires a ce sujet viendront sous peu.

Exemple de sortie

Qbkysjdi Doxacyd m mb zmsj igesaytwp. Zvww mvtqy ksmivj lzt fpepqdzrua joypzkjapljngsu bmbnimima msxph ysmwhj mg spfte qtwfukn. Hpjromny, qent qnosf gkf jtauwgh qqt us dykfneiip, zvww lowjmsu q zozf glsqyew fie vdg pqdzrua gnabtvodmny. Am taubqp mntwa qz,bp yvkysj x mqpkt pwpyroig dj oar tot lf wpfoade xlt mbeieqkg spoy ntly ksmivtqy. Gf, qdtnitr dqjiydasf ri vfbyhfqi Isoaluq, u,ohjiyf renh mg efvmrjc pr wqasxpmn qazdfqd bva p,evlayf rg cmptliairx cglplde ipy rdbmnybpmppy ptlt kfddew mg lpdliwng po b,issaenat cjidr qmysngamjfg. Isseenqditqy hvvjljngsuqarjf m lf wijji ndfdbrtqd hpoy pj.

Explication du début de la sortie

On doit d'abord corriger la date d'envoi de 3185/11/25 à 3195/11/35, ce qui nous donne la clé [3, 1, 9, 5, 1, 1, 3, 5]. La date de réception nous donne [3, 1, 8, 8, 0, 5, 1, 8]. On commence ensuite à progresser dans le message. $3 \times 22 (V) + 3 = 69$. $69 \bmod 26 = 17$, ce qui correspond à Q. On parcourt ensuite le message avec la clé. $1 \times 1 (a) + 1 = 2$, ce qui correspond à b (déjà en mod26). Après le premier mot, on saute l'espace et on arrive au début du deuxième mot. C'est la 9^e lettre du message, mais notre clé ne comporte que 8 chiffres, ce qui fait qu'on recommence la clé au début. Ainsi, $3 \times 9 (l) + 3 = 30$. $30 \bmod 26 = 4$, ce qui correspond à D. On

continue ainsi en esquivant tout ce qui n'est pas une lettre et en ne comptant pas ces caractères spéciaux dans notre encodage.

Problème #15 - Calculateur de Nombre Complexes (15 points)

Ayant à peu près fini de réparer tout le vaisseau, vous décidez de vous reposer en explorant les nombres complexes. Pendant votre "pause", vous devrez donc coder un programme qui saura reconnaître les nombres complexes et les opérateurs de base (+ − × ÷). Pas de panique ! Vous n'aurez pas besoin de coder les exposants, les racines et la priorité des opérations. Votre pause n'est pas si longue quand même...

Pour ce qui est des nombres complexes, ils prennent généralement la forme $a + bi$, mais peuvent aussi s'écrire a si $b = 0$, par exemple. a et b sont des coefficients réels (ne pas oublier qu'ils peuvent être négatifs) et i est le nombre imaginaire $\sqrt{-1}$ (qui n'existe pas dans les réels). On dit donc que a est la partie réelle d'un nombre complexe noté z , et que bi est la partie imaginaire. Le truc principal à savoir avec les nombres complexes, c'est que $i \times i = -1$. Sachant cela, on peut se lancer dans les opérations élémentaires...

Pour l'addition et la soustraction, c'est simple : on additionne ou soustrait les parties réelles des deux nombres complexes ensemble et il en va de même pour les parties imaginaires.

$$\begin{aligned} z_1 \pm z_2 \\ &= (a + bi) \pm (c + di) \\ &= (a \pm c) + (b \pm d)i \end{aligned}$$

où $a \pm c$ est la partie réelle du complexe z résultant et $(b \pm d)i$ est sa partie imaginaire.

Pour la multiplication, **qui sera noté par * dans les opérations**, il faut faire toutes les multiplications possibles, comme si on avait $(a + bx) * (c + dx)$.

$$\begin{aligned} z_1 * z_2 \\ &= (a + bi) * (c + di) \\ &= ac + adi + bci + bdi^2 \end{aligned}$$

Le reste du développement vous appartient mais, ce qui fait que la multiplication est aussi simple, est que i^2 devient -1 , laissant bd réel (mais modifié par un $-$). On se sort donc de la multiplication avec, encore une fois, un nombre complexe de la forme $a + bi$.

Pour l'opération la plus **complexe**, soit la division (**qui sera noté par / dans les opérations**), il faut d'abord introduire la notion de conjugué complexe. Pour tout nombre complexe z , il existe un conjugué complexe \bar{z} tel que, si $z = a + bi$, alors $\bar{z} = a - bi$. L'avantage de la multiplication par le conjugué complexe est que le nombre résultant est réel car le terme en i s'annule et le terme en i^2 devient réel. Ainsi, lorsque nous effectuerons la division d'un nombre complexe par un autre, nous devrons multiplier le diviseur et le dividende (pour pas que ça ait un impact sur le résultat) par le conjugué complexe du diviseur, nous laissant un nombre complexe en haut et un réel en bas. Le réel se distribue

ensuite sur chacune des deux parties du nombre complexe pour réduire son coefficient. On procède donc comme suit :

$$\begin{aligned} & \frac{z_1}{z_2} \\ &= \frac{z_1 * \underline{z_2}}{z_2 * \underline{z_2}} \\ &= \frac{(a + bi) * (c - di)}{(c + di) * (c - di)} \end{aligned}$$

où on obtient $c^2 - d^2i^2$ en bas. Le dénominateur de la fraction devient donc réel. Le reste du développement au dénominateur et le développement du numérateur vous est laissé pour votre plaisir personnel.

Spécification d'entrée

La première ligne correspond au nombre de tests à effectuer. Les lignes suivantes correspondent chacune à suite d'opérations précises à effectuer de gauche à droite (les opérateurs seront utilisés de sorte qu'il n'y ait pas de problème de priorité des opérations). Il y aura un espace entre tous les nombres complexes et les opérateurs, et il n'y aura pas de parenthèses encadrant les nombres complexes par souci de simplicité. $((a + bi)$ sera noté $a + bi$)

Spécification de sortie

Vous devrez sortir le nombre complexe résultant de la chaîne d'opérations sous la forme $a + bi$. Attention ! Si $b = 0$, vous devrez afficher seulement a et vice-versa si $a = 0$ (bi). Si les deux valent 0, votre programme doit bien afficher 0 et non pas rien. Si les coefficients a et/ou b ne sont pas entiers, ils seront arrondis au centième le plus près (donc des nombres à virgules).

Exemple d'entrée

```
4
5+7i + 2-3i - -8+19i
6-2i * -7-4i * 9-13i - 1+6i
-5-8i / 1-i
13+25i / -3+2i / 58-i * 3+7i / 5-2i * 7+9i - 8+8i + -3+2i - 5+6i
```

Exemple de sortie

```
15-15i
-581+554i
1.5-6.5i
-14.85-10.16i
```

Explication de la première sortie

On a affaire à des additions et soustractions de nombre complexes, donc on sépare le nombre en parties réelle et imaginaire. Dans les réels, on a $5+2-8 = 15$ et dans les imaginaires, on a $7-3-19 = -15$. On obtient donc $15-15i$.