



CRC
**COMPÉTITION DE
PROGRAMMATION**



**PROBLÈME
PRÉLIMINAIRE 1**

QUELQUES NOTES

- Les règles complètes sont dans la section 4 du livret des règlements
- Vous avez jusqu'au **Dimanche 3 décembre, 11:59 pm** pour remettre votre code
- N'hésitez pas à utiliser le forum de programmation sur le discord de la CRC pour poser vos questions et discuter des problèmes. Il est là pour ça!
- **On vous donne des fichiers modèles faciles à utiliser pour votre code et pour faire vos tests. Merci de les utiliser.**

UTILISATION DU FICHIER MODÈLE

- En gros, le fichier de test appelle la fonction associée avec en paramètre les informations du test et compare sa sortie avec ce qui est attendu pour vous permettre de voir si ça fonctionne. **Tout votre code (sauf fonctions additionnelles et classes que vous avez créé, qui devraient aller juste au-dessus) devrait être écrit dans la fonction prévue à cet effet.**
- Les points mis dans le document sont indicatifs de la difficulté de chaque section et du pointage pour leur réussite pour ce défi. Ce défi préliminaire va avoir une valeur globale de 2% du défi principal. Pour plus d'informations, aller voir la section 4 du livret des règlements.

STRUCTURE

Chaque problème contient une petite mise en situation comme celle-ci expliquant les fondements du problème et donnant les bases nécessaires pour résoudre celui-ci. Chaque problème préparatoire contient aussi la répartition des points pour le problème.

Spécification d'entrée et de sortie:

Dans cette section, on retrouve les caractéristiques des entrées qui peuvent être fournies au code en question ainsi que les critères attendus pour les sorties du programme.

Exemple d'entrée et de sortie:

Dans cette section se trouve un exemple d'entrée (parfois constitué lui-même de plusieurs sous-exemples) pour que vous puissiez tester votre code. L'exemple de sortie donne donc la réponse attendue pour cette entrée.

Explication de la première sortie:

Si le problème n'est toujours pas clair après la mise en situation, l'explication de la première sortie sert parfois à démêler le tout en expliquant comment la première entrée est traitée et en montrant le chemin menant à cette réponse.

C'est cryptique tout ça!

Partie 1: Trouvez le code secret (4 points)

Vous avez trouvé un document crypté dans votre ordinateur et vous souhaitez le déchiffrer. Vous savez que ce document contient la clé nécessaire pour le déchiffrer. Vous devrez donc trouver la clé. Pour aider votre programme à trouver la clé, voici un indice: la clé est un nombre qui se trouve entre deux symboles "|" (124 en ascii base 10 ou 7C en base HEX). Ce nombre est la clé pour le décryptage du fichier!

Spécification d'entrée:

En entrée, vous recevrez un message crypté dont les mots n'ont aucun sens. Dans ce message se cachera le nombre à trouver entre deux caractères "|".

Spécification de sortie:

En sortie, vous devrez renvoyer la clé trouvée dans le texte. Cette clé est le nombre contenu entre les caractères "|".

Exemples d'entrée:

```
ti kzk mab cvm kwuxmbqbgwv bzma|18| ncv
```

```
Zcfsa wdgia rczcf gwh oash, qcbgsqhshif orwdwgqwbz szwh. Qfog qifgig  
dzoqsfoh hcfhcf bsq toqwzwgwg. Bizzoa jsz sfoh zsc. Gigdsbrwggs bcb  
bwgz sbwa. Aosqsbog oq sfoh sush sfoh gcrozsg hsadig. Bizzo toqwzwgw.  
Dszzsbhsgeis toqwzwgwg|12| hwbqwribh bibq, dfshwia hwbqwribh rczcf  
jsbsbohgw wb. Bizzoa woqizwg oqqiagob zsc bcb sttwqwhif. Aosqsbog  
crwc ofqi, hfwghweis.
```

Exemples de sortie:

18

12

Explication de la première sortie:

Dans le premier message, nous avons, vers la fin du texte, la séquence "|18|". Cette séquence est la clé que nous cherchons à découvrir. Il nous reste maintenant à en extraire le nombre et à le renvoyer en sortie.

Partie 2: Décryptons César (8 points)

Une fois que vous avez la clé, vous pouvez maintenant décoder le fichier. Le fichier est encodé avec un *Caesar cipher*. Le *Caesar cipher* est une encryption par décalage des lettres. En effet, on prend les lettres de l'alphabet et on les décale vers la droite d'un nombre spécifique, nombre qui est la clé de décryptage. Ainsi, si la clé de décryptage est 3, a devient d, b devient e, c devient f et ainsi de suite jusqu'à y devient b et z devient c. Vous devez donc décaler le message selon la clé donnée. Tous les symboles de la phrase qui ne sont pas des lettres ne doivent pas être modifiés!

Spécification d'entrée:

En entrée, vous recevrez un message crypté dont les mots n'ont aucun sens et une clé de décryptage, soit un nombre entre -26 et 26.

Spécification de sortie:

En sortie, vous devrez retourner le message décalé selon la clé fournie.

Exemples d'entrée:

```
ti kzk mab cvm kwuxmbqbwv bzma ncv!
```

18

```
Zcfsa wdgia rczcf gwh oash, qcbgsqhshif orwdwgqwbz szwh. Qfog qifgig  
dzoqsfoh hcfhcf bsq toqzwzgw. Bizzoa jsz sfzh zsc. Gigsbrwgs bcb  
bwgz sbwa. Aosqsbog oq sfzh sush sfzh gcrozsg hsadig. Bizzo toqzwzgw.  
Dszzsbhsgeis toqzwzgw hwbqwribh bibq, dfshwia hwbqwribh rczcf  
jsbsbohgw wb. Bizzoa woqizwg oqqiagob zsc bcb sttwqwhif. Aosqsbog  
crwc ofqi, hfwghweis.
```

12

Exemples de sortie:

```
la crc est une competition tres fun!
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras cursus  
placerat tortor nec facilisis. Nullam vel erat leo. Suspendisse non  
nisl enim. Maecenas ac erat eget erat sodales tempus. Nulla facilisi.  
Pellentesque facilisis tincidunt nunc, pretium tincidunt dolor  
venenatis in. Nullam iaculis accumsan leo non efficitur. Maecenas  
odio arcu, tristique.
```

Explication de la première sortie:

Dans la première sortie, on trouve la clé de décryptage qui est 18. On va maintenant prendre chaque lettre dans le texte et la décaler de 18 dans l'alphabet pour retrouver le message initial. Notez bien que la ponctuation n'a pas été modifiée de la phrase initiale.

Partie 3: Perdu dans une mer de mots (8 points)

Le texte décrypté ne semble avoir aucun sens particulier et être juste un amas de mots. Le vrai message est noyé dans un océan de mots aléatoires. Vous devez trouver les 5 mots les plus fréquents du texte et les ordonner du plus fréquent au moins fréquent. Ces cinq mots forment le message que vous recherchez!

Spécification d'entrée:

En entrée, vous recevrez un message de mots aux fréquences variées.

Spécification de sortie:

En sortie, vous devrez retourner les 5 mots les plus fréquents en commençant par le plus fréquent.

Exemples d'entrée:

chien le chat maison voiture message arbre secret soleil fleurs
musique rive montagne est ordinateur plage le cell lune livre mer
secret ceci restaurant amis le voyage chanson bateau hiver automne
neige amour sport bonheur le art danse message avion jardin livre
parc est cadeau table chapeau porte lunettes secret chemin message
commande fort pont or argent fourchette message couteau tasse montre
chapeau guitare le piano balance le nuage parapluie est balle
chocolat ceci message chaussettes radio silence plume pinceau secret
tableau nuage le bougie chemin miroir porte montre papier le lampe
sac message bonjour le sable

chocolat la pluie est la plage est pomme la sapin la musique fun
vraiment ordinateur prog prog cauchemar la montagne bonjour arbre
prog amour livre vraiment prog couleur danse guitare voyage chanson
est horizon fleur fun prog livre lune ballon vraiment ami prince la
plume prog vent porte la table aventure la neige est chaussure
histoire bateau

Exemples de sortie:

le message secret est ceci

la prog est vraiment fun

Explication de la première sortie:

On prend les fréquences de chaque mot et le mot "le" est celui qui revient le plus souvent. Le prochain mot qui est le plus fréquent est "message". On continue jusqu'à trouver nos 5 mots les plus fréquents, ce qui donne: "le message secret est ceci".

Partie 4: Cascade de mots (12 points)

Pour afficher le message de manière plus divertissante nous allons les organiser en une belle cascade. Pour ce faire, nous allons prendre 5 mots et les imprimer en cascade. Les mots sont disposés de la manière suivante: chaque lettre d'un mot se trouve sur la ligne d'en dessous et un caractère de plus vers la droite. Entre chaque mot se trouve un espace pour les séparer. Veuillez noter que si un mot est plus court que les autres, ses lettres sont remplacées par des espaces.

Spécification d'entrée:

En entrée, vous recevrez une string contenant 5 mots.

Spécification de sortie:

En sortie, vous devrez retourner une liste de string qui contient tous les étages de la cascade l'un après l'autre.

Exemples d'entrée:

```
voici un petit texte test
```

```
la prog est vraiment fun
```

Exemples de sortie:

```
[
"v u p t t",
" o n e e e",
"  i   t x s",
"   c    i t t",
"    i     t e "]

[
"l p e v f",
" a r s r u",
"   o t a n",
"    g   i ",
"     m  ",
"      e ",
"       n ",
"        t "]
```

Explication de la première sortie:

On prend chacun des mots et on les fait en étage en mettant des espaces entre les mots. Affichés l'un à la suite de l'autre, ces étages forment une cascade.

```
v u p t t
 o n e e e
  i   t x s
   c    i t t
    i     t e
```

Partie 5: Pouvez-vous faire mieux? (8 points)

Faites votre propre sortie de texte intéressante comme dans la **partie 4** avec 5 mots aléatoires. Vous devez donc faire votre propre sortie intéressante et coder la solution. En plus du code, vous devez écrire trois tests vous-même avec pytest pour vérifier que votre modification du texte fonctionne bien. (Nous allons regarder le code et le test)

Spécification d'entrée:

En entrée, vous recevrez une string contenant 5 mots.

Spécification de sortie:

C'est à vous de décider!

Exemple d'entrée:

```
voici un petit texte test
```

Exemples de sortie:

```
???
```

Explication de la première sortie:

Soyez créatifs!