Problemes d'Algorithmie

Astremy

Table des matières

Exercices

Facile

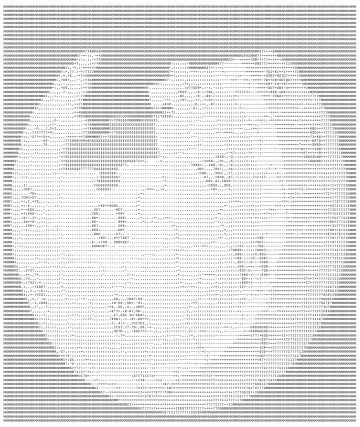
Problème Numéro 1

Vous trouvez, en vous perdant sur le net, un jeu bien sympathique :
Jouer aux 7 différence sur des ascii-art.
Au bout d'un moment, l'envie vous prend de tricher un peu, en mettant au point un programme permettant de détecter automatiquement les caractères différents.
Cela se fait signaler par le programme en remplaçant le caractère en question par un "x".

Exemple :

\(OwO)/
et
\(Owo)/
=>
\(Owx)/

Note : Dans les exemples, on voit bien la différence. Sur de grand ascii-art cela deviens bien moins évident.



Problème Numéro 2

Vous découvrez le principe d'asservissement :
C'est le fait que un système récupère des informations sur les conséquences de ses actions pour s'auto-réguler.
Votre programme sera le systeme, vous répondrez a la manière de capteurs, pour dire si c'est trop ou trop peu.
Au début du programme, vous utilisez une entrée utilisateur (input) pour dire deux nombre entre lesquels vont être compris votre nombre, et choississez mentalement une valeur dedans.
A chaque fois, votre programme devra vous faire une proposition, et vous

devrez dire si le nombre est plus petit, plus grand ou égal.

Exemple :
Le 0 indique l'utilisateur, et le 1 le programme
(0) 0 10 (le nombre est compris entre 0 et 10, 0 et 10 non compris)
(1) 5 ?
(0) +
(1) 7 ?
(0) +
(1) 8 ?
(0) =
(1) Yay !

Problème Numéro 3

Vous êtes sur twitter, et rencontrez un problème assez.. problématique.. Le tweet que vous souhaitez envoyer fait plus de 200 caractère, vous ne pouvez donc pas envoyer le message.
Heuresement, vous avez une technique : Enlever le plus d'espaces inutiles.
Par espaces inutiles, voue entendez les espaces avant ou après les virgules, les points (tout les types), et les parenthèses.
Cela vous permet de gagner quelques caractères, qui font parfois la différence.

Exemple :
Bonjour à tous, je suis ici aujourd'hui pour vous parler d'un projet (sur lequel je travaille depuis un bon bout de temps) !
=>
Bonjour à tous,je suis ici aujourd'hui pour vous parler d'un projet(sur lequel je travaille depuis un bon bout de temps) !

Exemple 2 :
Salut les amis ! Je m'appelle Lucie ! Connaissez vous piscord ? Si vous ne savez pas, je serait heureuse de vous en parler..
=>
Salut les amis !Je m'appelle Lucie !Connaissez vous piscord ?Si vous ne savez pas,je serait heureuse de vous en parler..

Faire un programme pour automatiquement simplifier un tweet.

Problème Numéro 4

Lors d'une partie de scrabble, vous vous dites que trouver plus facilement des mots avec vos lettres peut être un réel avantage par rapport à votre adversaire.
Vous décidez donc de faire un programme qui, avec une liste de mots et les lettres que l'on as, renvoie tout les mots que l'on peut faire.

Exemple :
aelkims
hello je fait le nya why kems
=>
le kems

Exemple 2 :
uiheany
hello je euh le nya why kems
=>
euh nya

Problème Numéro 5

Le markdown est un langage de balisage, qui est utilisé par beaucoup de choses dans le monde de la programmation, notamment github, avec les fichiers de présentation.
Vous décidez de refaire une fonctionnalité de markdown, mais à votre sauce : Les grands textes.
Ils sont, en markdown, qualifié par la balise "#", qui signifie un titre, "##" pour un sous-titre, "###" pour un sous-sous-titre, et ainsi de suite.
Seulement, les titres ne sont pas assez gros pour vous ! Ainsi, au lieu de faire que cela descende la taille du texte, plus il y aura de #, plus il sera gros, et cela à l'infini.
Avant de créer votre langage, vous souhaitez faire un shéma pour montrer à quoi ça va ressembler.
Cela donne cela :

""""
Titre 1
Titre plus gros !
Bonjour à tous
Comment # allez vous ?
Moi # ça va
Nyaaa
""""
=>
""""
[Gros|Titre 1]
[Très gros|Titre plus gros !]
Bonjour à tous
Comment [Gros|allez vous ?]
[Très gros|Moi [Gros|ça va]]
[Très très très gros|Nyaaa]
""""

Tache : Faire un programme qui, à partir d'une entrée, en sort le shéma

de démonstration montrant la taille du texte.

Problème Numéro 6

Suite à un sondage qu'à fait graven, vous cherchez à récolter les notes pour calculer la moyenne et la médianne.
Seulement, les messages de chacuns sont long, et il y en a beaucoup. Vous remarquez que la note est toujours avant un /, et que ensuite c'est sur combien elle est (20, 50, 100...).
La note finale doit être une moyenne et une médiane en pourcentage (arrondis)

Exemple :
""""
Bon finalement 17/20 car Lazor et Horus ne sont pas la et je me suis trompé mais Pim nous a quitté

10/20
Trop d'histoires, les modos sont trop gentils.
""""
=>
Moyenne : 68%
Mediane : 68%

Problème Numéro 7

Dans un temple secret de la jungle amazonienne (me demandez pas comment vous avez fait pour arriver là) vous trouvez un pattern de déplacement pour trouver une porte secrète.
Seulement, il faut répéter le pattern un nombre spécifique de fois, mais vous ne savez pas combien de fois. Pour trouver cette fréquence, vous trouvez dans la pierre une suite qui se répète. L'espace entre chaque symbole correspond au nombre de fois qu'il faut faire le pattern !
Vous y êtes presque, mais avez un dernier probleme qui se pose : Les signes ne sont pas bien lisibles, vous ne voyez que certains symboles.
Trouvez la fréquence du symbole qui se répète.

Exemple :
A ? ? ?A ?A ?A ? ? ?A
=>
2

On obtient 2 car la répétition est surement A ?A ?A ?A ?A ?A ?A

Problème Numéro 8

Vous reprenez twitter, mais remarquez toujours le même probleme : La taille limite des tweets.
Cette fois, vous avez un message beaucoup trop gros, et n'allez pas essayer d'esquiver le probleme en enlevant des espaces.
Vous allez l'envoyez en plusieurs fois. Seulement, vous ne souhaitez pas couper n'importe ou votre message, vous souhaitez le couper sur des espaces, à chaque fois, et décidez de faire des messages avec l'espace le plus proche des 200 chars.

Exemple :
Bonjour à tous, je m'appelle Lucie, et je fait les exercices du jour. Est-ce que vous vous faites des exemples tout les jours de phrases, de situations, tout ce genre de truc. A force cela deviens fatiguant, je vous le dit ! Un exemple, ça va. Cinq exemples, passe. Cinq tout les jours pendant un mois c'est compliqué.
=>
2 (nombre de messages)
Bonjour à tous, je m'appelle Lucie, et je fait les exercices du jour. Est-ce que vous vous faites des exemples tout les jours de phrases, de situations, tout ce genre de truc. A force cela deviens

fatiguant, je vous le dit ! Un exemple, ça va. Cinq exemples, passe. Cinq tout les jours pendant un mois c'est compliqué.

Problème Numéro 9

Vous vous amusez à utiliser des cercles pour faire plusieurs formes géométrique sur votre cahier. Vous pouvez en faire des carrés, en faisant un bloc de cercles, ou en triangle en mettant un cercle, puis deux, puis trois, et ainsi de suite.
Après avoir fait vos dessin, vous vous amusez à calculer combien de cercles comprennent chaque forme.
Vous remarquez que un triangle de 8 de hauteurs à exactement le même nombre de cercles qui le composent qu'un carré de 6 de coté.
Il vous viens une question en tête : Est-il possible de, étant donné un nombre, déterminer s'il est possible de faire un carré avec, un triangle, les deux ou aucun des deux ?
Vous décidez de faire un programme pour tenter de répondre à cette question.

Exemple :
36
=>
36 est un nombre carré et triangulaire

Problème Numéro 11

Vous regardez dans votre porte-monnaie, et remarquez que vous avez quand même énormément de centimes.
Vous pourriez simplement transformer tout les centimes seuls en 2 centimes et diviser par 2 le nombre de ces derniers. En poursuivant le raisonnement, vous pourriez convertir en 5 centimes, et ainsi de suite.
Etant donné un nombre donné d'euros, donner la composition de pieces/billet différent maximale que l'on peut faire (en supposant qu'il y a des billets d'une taille infinie.)

Exemple :
32579854.31
=>
20000000, 10000000, 2000000, 500000, 50000, 20000, 5000, 2000, 2000, 500, 200, 100, 50, 2, 2, 0.2, 0.1, 0.01

Problème Numéro 12

Vous lisez quelques citations célèbres. Au fur et à mesure des lectures, vous remarquez des choses intéressantes.
Souvent, les points fort d'un texte sont les mots les plus longs, ou les mots contenant une ou plusieurs majuscules. Si une phrase contient un point d'exclamation, la phrase entière est importante.
En effet, en premier point, les majuscules. Cela sert à exprimer une idée forte "I HAVE A DREAM" (même si peu transféré dans la version écrite, on imagine que c'est le cas). Ensuite les points d'exclamation, qui montre une insistance "Les gens ne sont pas dupes !". Enfin, les mots long, permettent de présenter souvent des adjectifs qui vont aider à décrire la situation ("infiniment", "malheuresement"..)
Etant donné une phrase, en sortir les points forts.

Moyen

Problème Numéro 1

Vous étudiez la luminosité des cases dans un célèbre jeu : Minecraft
En posant une torche, vous remarquez que la façon dont la lumière se met est en forme de losange :
En effet, un bloc est éclairé à 1 si un bloc a coté de lui est éclairé à 2, qui est éclairé à 2 si un bloc a coté est éclairé à 3, et ainsi de suite.
Cela se représente comme ça, pour une torche qui éclaire par exemple à 3 :

0 0 0 0 0 0 0
0 0 0 1 0 0 0
0 0 1 2 1 0 0
0 1 2 3 2 1 0
0 0 1 2 1 0 0

0 0 0 1 0 0 0
0 0 0 0 0 0 0

La valeur 3 représente la torche, qui éclaire donc le bloc d'a coté de 2, ainsi de suite.
Etant donné un éclairage donné de la torche, déterminer les coordonnées ou il y a de l'éclairage, la torche se trouvant toujours en 0,0.

Exemple :
2
=>
1,0 0,1 0,0 0,-1 -1,0

Problème Numéro 2

Vous rangez votre bibiliothèque, mais avez de petites manies de rangement :
Pour commencez, vous triez les livres selon leur genre (policier, fantastique,conte..), pour savoir dans quelle armoire vous la mettez. Le premier genre que vous rencontrez va être le genre de l'armoire numéro 1, le 2eme celui de l'armoire numéro 2 et ainsi de suite.
Si vous rencontrez deux fois le même genre, ils irons tout les deux dans la même armoire, il n'y a pas une armoire par livre.
Ensuite, vous les triez alphabétiquement par auteur, et si plusieurs livres ont le même auteur, alphabétiquement par titre, indépendamment des majuscules ou non.
On vous donne une liste de livre avec Genre Auteur et Titre, et vous devez donner l'armoire et la place dedans de l'oeuvre demandée après.

Exemple :
3 (Le nombre de livres)
Roman / Marie-Madeleine de La Fayette / La princesse de clèves
Roman d'anticipation / Jules Vernes / De la terre a la lune
Roman d'anticipation / Aldous Huxley / Le meilleur des mondes
.
Roman d'anticipation / Jules Vernes / De la terre a la lune (On demande un livre)
=>
Armoire n°2, Position 2

Problème Numéro 3

Vous essayez de faire une calculatrice simple. Vous essayez déjà de traiter une simple opération.
Pour cela, vous faites quelque chose de la sorte : nombre A (opérateur) nombre B (opérateur) nombre C.
Le but est de calculer le résultat de cela, en gardant les priorités opératoires,

par exemple 3+4/2 donnerais 5 (4/2 = 2, 3+2 = 5).
Pour vous simplifier la tache, il n'y a pas de calcul qui renvoie un nombre a virgule.
Information importante : Vous ne devez rien faire pour calculer automatiquement, vous devez "parser" le calcul vous-même (Sinon, le problème ne servirais a rien).

Exemple :
4*3-1
=>
11

Problème Numéro 4

Vous vous battez contre des ennemis dans un jeu de style RPG. Vous tombez contre plusieurs ennemis a la fois, mais ne savez pas si vous devez plutôt commencer par attaquer les petits, les gros..
Vous vous demandez comment savoir l'ordre dans lequel attaquer les ennemis.
Le jeu est en tour par tour : Chaque tour, vous attaquez un ennemi, puis tout les autres ennemis vous attaquent.
Soit donné : Une attaque, un nombre d'ennemis et après l'attaque et la vie de chaque ennemi, donnez l'odre dans lequel on doit les attaquer pour prendre le moins de dégats possibles.

Exemple :
10 (attaque)
3 (nombres d'ennemis)
100 10 (Premier ennemi : 100 d'attaque et 10 de vie)
10 100 (Deuxième ennemi : 10 d'attaque et 100 de vie)
30 50 (Troisième ennemi : 30 d'attaque et 50 de vie)
=>
1 3 2 (Premier ennemi à attaquer : 1, puis 3, puis 2)

Problème Numéro 5

Vous découvrez que des personnes sont capables de trouver quel jour de la semaine (lundi mardi..) sera n'importe quel jour de n'importe quelle année !
Impressionné, vous décidez d'essayer de faire quelque-chose de similaire, mais en commençant petit.
Essayez de faire que, soit un jour donné, donner le jour suivant.

Exemple :
3/5/2020
=>
4/5/2020

Exemple 2 :
31/7/2020
=>
1/8/2020

N'utilisez pas de librairie ou truc qui est déjà fait pour ça !

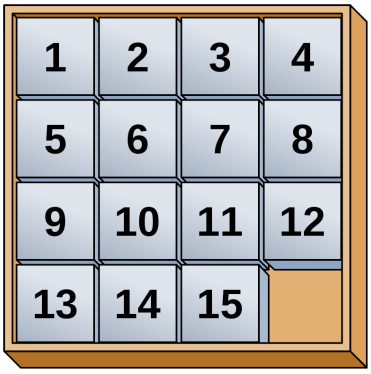
Problème Numéro 6

Vous vous regardez dans un miroir, et remarquez que le monde dedans est à l'envers. Vous approchez votre carnet, et constatez que vous n'arrivez pas à lire l'écriture.
Soit un texte, donner le rendu que ce dernier a dans le miroir (symétrique).

Exemple :
""""
pp ><
HH AA
""""
=>
><qq
HH AA

Problème Numéro 7

Le taquin est un jeu se présentant par une grille de 4 par 4 ou l'on peut faire glisser les pièces, le but est de le résoudre pour qu'il prenne la position selon laquelle les pieces sont ordonnées avec le trou à la fin.



Seulement, la moitié des configurations possibles du taquin sont impossible à résoudre.
Avec les ressources, faire un programme pour déterminer si une configuration jeu est résoluble (0 correspond à la case vide)

Exemple :
1 2 3 4

5 6 7 8
9 10 11 12
13 15 14 0
=>
Non résoluble

Ressources :
Video de Micmaths
Wikipedia

Problème Numéro 8

La loi de Godwin est une règle empirique énoncée en 1990 par Mike Godwin, d'abord relative au réseau Usenet, puis étendue à l'Internet :

« Plus une discussion en ligne dure, plus la probabilité d'y trouver une comparaison impliquant les nazis ou Adolf Hitler s'approche de un. »
Vous décidez de mettre cela en oeuvre, en recherchant un point Godwin dans les pages de sites connus.
Etant donné un lien url original, utilisez les href de cette page pour naviguer jusqu'à trouver le mot "nazis" ou "hitler".

Exemple :
https ://fr.wikipedia.org/wiki/Loi_de_Godwin
=>
Point Godwin : https ://fr.wikipedia.org/wiki/Loi_de_Godwin
En effet, la page contient un des mots, ce qui fait qu'elle n'à pas besoin d'aller plus loin.

Problème Numéro 9

Dans votre livre "la chimie pour les nuls", vous voyez diverses équations chimiques, par exemple "C + O2 → CO2". Cela veut dire "un atome de carbone (C) + 2 atomes d'oxygène (O) font une molécule de CO2 (dioxyde de carbone)".
Ps : O2 est en faite une molécule de dioxygène, mais je vais éviter de compliquer les choses.
L'essentiel pour vérifier qu'une équation chimique soit valide est qu'il y ait autant de composants de chaque coté. Dans ce cas, 2 atomes d'oxygène et un atome de carbone.
Autre exemple d'équation : CH4 + 2 O2 → 2 H2O + CO2
Quand il y a un nombre orphelin devant une molécule (une molécule est une liaison de plusieurs atomes, comme par exemple "CO2"), cela signifie que la molécule est en multiple quantité.
Par exemple, "2 H2O" signifie qu'il y à 2 molécules d'H2O, donc 4 atomes d'hydrogène (H) et 2 atomes d'oxygène.
Faire un programme qui vérifie si une équation est valide.

Exemple :
CH4 + O2 >H2O + CO2
=>
Non valide

En effet, ce n'est pas une équation valide, en effet, d'un coté il y a 4 atomes d'hydrogène, de l'autre 2, et d'un coté 2 d'oxygène mais de l'autre 3.

Problème Numéro 10

Une queue est une suite d'éléments, avec une capacité simple : Soit l'on peut lire le premier élement de la queue, soit l'on peut ajouter un élement à la fin de la queue.
Vous essayez d'implémenter cela pour que la complexité de n'importe laquelle des 2 opérations sur la queue se fasse en O(1), ainsi qu'avoir la longueur, ou les choses comme ça.

Problème Numéro 11

Vous essayez de convertir un nombre n de secondes en heures, minutes, et secondes. Après un peu de reflexion, vous vous dites qu'utiliser une base 60 serait utile pour cela.
Au lieu de faire simplement ça, vous décidez de faire un convetisseur d'un nombre décimal en une autre unité, pour n'importe quelle base choisie entre 2 et 62 (62 parce que 0 à 9 + 26 lettres majuscules + 26 lettres minuscules).

Exemple :
15 (base)
43 (nombre)
=>
2D

Problème Numéro 12

Vous regardez votre feuille.. Elle est pleine de calculs. Faire un calcul est extrèmement facile pour un programme..
Nombre + Nombre renvoie directement le résultat, c'est si facile..
Et si vous avez des chaines de caractères, vous avez juste à le transformer en nombre pour l'utiliser.
Une idée un peu étrange vous viens cependant en tête : Comment feriez-vous si vous ne pouviez pas transformer la chaine en nombre ?
Essayez de réaliser un programme qui fait une addition, en utilisant seulement des chaines de caractère, en ne passant pas par des nombres.
Tentez en binaire pour commencer, ensuite voyez si vous pouvez améliorer.

Exemple :
1010110011 (première chaine de caractère)

1100101011 (deuxième chaine de caractère)
=>
10111011110

Difficile

Problème Numéro 1

Un jour vous prend l'envie de faire de la stéganographie (l'art de dissimuler des choses), et de cacher du texte dans une image. Pour ce faire, après de nombreuses recherches internet, vous tombez sur une bonne technique :
Lire les données de l'image, et utiliser le RGB :
Pour chaque pixel de l'image, utiliser RG et B pour mettre les valeurs au bit le plus proche, une valeur étant un bit.
En gros, si le bit de la donnée est sensée être un 1, le faire que la valeur modulo 2 soit 1, exemple, 132 doit être passé à 131 ou 133.
Cela est fait pour RGB, ainsi, nous avons 3 bits codés par pixel.
Tout les 3 pixels, la première valeur code la "validation", qui permet de vérifier que le texte ne continue pas. S'il est à 1 le texte continue, sinon, il s'arrête.

Nous avons ainsi tout les 3 pixels 9 valeurs : 1 de validation et 8 de valeur.
Ainsi, 3 pixels codent un octet de données.
Nous ainsi ainsi le texte à dissimuler, et prenons sa valeur ascii pour stocker un char sur 3 pixels.

Exemple :
Texte à dissimuler : "a"(juste le char, pour l'exemple)
=>Binaire a dissimuler : 1(validateur)01100001
On va dire que l'image est constitué juste de 4 pixels de cyan (12,182,249)
12 182 249 12 182 249 12 182 249 12 182 249

---

1 0 1 1 0 0 0 0 1 0 Le reste n'importe pas

---

13 182 249 13 182 248 12 182 249 12 182 249

Ainsi, les changement de l'image ne sont pas perceptibles, et nous pouvons y cacher un texte !
Essayez de faire un systeme de chiffrement et déchiffrement d'image ! (Si vous utiliser python, il y à PIL pour traiter les images)

Problème Numéro 2

Maintenant que vous avez fait de la stéganographie, il est temps de vous attaquer à la cryptographie !
Vous choisissez d'utiliser une méthode assez simple :

L'utilisateur choisi un nombre entre 0 et 25. Ensuite, la première lettre du message est prise, et décalée d'un certain nombre de places dans l'alphabet, suivant justement le nombre choisi.
Pour chaque lettre qui suit, on décale leur place du nombre de cases correspondant à la place dans l'alphabet qu'avait la lettre précédente.

Exemple :
salut (Message à chiffrer)
5 (Clé de chiffrement)

Traitement :
s (19eme lettre) + 5 = 24 =>x
a (1ere lettre) + s (19eme lettre) = 20 =>t
l (12eme lettre) + a (1ere lettre) = 13 =>m
u (21eme lettre) + l (12eme lettre) = 33. Cela dépasse de l'alphabet donc on repart du début = 6 =>g
t (20eme lettre) + u (21eme lettre) = 41. Cela dépasse de l'alphabet donc on repart du début = 15 =>o
=>
xtmgo

Ainsi, vous avez chiffré "salut" en "xtmfn" par exemple.
Voulant rendre votre message encore plus chiffré, vous décidez de transformer votre message en nombre, en utilisant la base 26.
Comment faire : Associez à chaque lettre sa place dans l'alphabet multiplié par la puissance de 26 correspondant à sa place dans le message - 1.

Par exemple, pour la première lettre du message et que c'est un x, on fait x, donc 24, multiplié par 26 puissance 1-1, 24 * 26**0 = 24
Pour la deuxième lettre, on a t (20) =>20 * 26**(2-1) = 20*26 = 520.

Au final, pour tout le message, nous avons 24 + 520 + 8788 + 123032 + 6854640 = 6987004. Le message sera chiffré par le nombre 6987004.

Essayer de faire un système pour chiffrer et déchiffrer un message avec ce système.
Info : Votre programme doit tout simplement supprimer les char autres que dans l'alphabet.

Problème Numéro 3

Vous devenez stagiaire chez une entreprise anglaise, et vous avez l'impression de n'être la que pour préparer du thé (on est en angleterre, c'est du thé, pas du café) à vos supérieurs toute la journée.
Un jour, votre patron vous le fait comprendre directement. Il vous demande d'implémenter un systeme de communication avec la théière pour pouvoir vous licencier mais continuer à avoir son thé.

Il vous explique que la RFC 2324 explique comment faire.
En vous appuyant sur cette dernière, mettez au point le protocole de communication avec la théière.

Problème numéro 4

Une requete web, c'est long à faire. C'est souvent dans un programme ce qui prend le plus de temps, car un ordinateur permet de faire plusieurs dizaines de milliers d'opérations par secondes.. Seulement, une requete à un temps fixe, impossible à changer, qui est beaucoup plus long que faire une opération.
Dans le projet de régler comme vous pouvez ce problème, vous commencer par constater quelque-chose : La requete est longue, mais ne sert parfois à rien. Si dans votre programme, vous n'utilisez pas la réponse de la requete, ce n'est en soit pas important que le programme attende la réponse. Votre idée est donc que au lieu de faire directement la requete, la tache soit ajoutée dans une queue et qu'une tache de fond se charge de la faire "quand elle aura le temps". Cela permet d'accélérer grandement le programme.
Seulement, une fois cela fait, vous remarquez que vous n'avez plus le moyen de récupérer la réponse de la requete.

Deuxième solution : Vous décidez d'au lieu de ne rien renvoyer et de ne pas pouvoir récupérer les réponses, de renvoyer un objet identifiant la requete dans la queue, et qui oblige le programme à attendre que la requête soit faite si l'on essaye de l'utiliser (ex : récupérer une propriété, appeller une méthode...)
Ainsi, le programme renvoie en soit de "faux objets" mais qui vont devenir des vrais quand la requête sera faite. Ainsi, si l'on n'utilise pas la réponse de la requête, elle se fera quand elle aura le temps, en parallèle. Par contre, si l'on doit utiliser la requête, on va utiliser le faux objet renvoyé qui va forcer la requête à se faire pour que l'on utilise le vrai résultat.

Essayez d'implémenter cela pour que ce soit invisible du point de vue utilisateur (ex : que l'on n'ai pas besoin de faire vrai_objet = faux_objet.wait(), que ce qui soit renvoyé soit ce que l'on utilise).

Pour aller plus loin :
Le fait d'attendre un certain temps de manière fixe est aussi une grosse perte de temps pour le programme : L'on pourrais faire autre chose en même temps. Essayez d'implémenter un système qui va, avec une listes de fonctions à exécuter, les faires, avec un système pour en lancer d'autres pendant que l'une "dort" et qui va revenir quand elle pourra après avoir fait les autres, ou que les autres dorment aussi.

Problème numéro 5

Une machine de Turing est une machine utilisant des données, faisant des opérations simplistes dessus, et en ressortant un résultat.
Pour transformer ses données, elle utilise 3 choses :
Un curseur, qui va permettre de lire la donnée sur un bloc, et le réécrire si necessaire. Il ne peut se déplacer que d'une fois sur la droite ou une fois sur la gauche après chaque opération.
La donnée lu sur le bloc ou est le pointeur.
L'état interne de la machine. La machine à plusieurs "états" différents. Selon la combinaison état interne + Donnée, on pourra déterminer ce qu'il faut faire.
Ce fut un concept qui aida en grande partie à développer un concept de l'informatique proche de celui que l'on connais aujourd'hui.
Aujourd'hui, nos ordinateurs, imprimantes, toutes les machines que l'on connais, sont des dérivés de machines de turing.

Une machine de turing "universelle" est une machine de turing permettant en principe de simuler n'importe quelle autre machine de turing, quelle que soit sa complexité, et permettant de faire la même chose, avec le même nombre d'opérations que la machine de turing qu'elle simule.
Un calcul ne sera long/compliqué pour la machine universelle uniquement et seulement si elle l'a été pour la machine de base.

Essayez de faire une machine universelle de turing (je vous le dit, vous aller en manger du wikipedia).

Problème numéro 6

Nous y voici !
Après avoir fait de la stéganographie et de la cryptographie symétrique, voici l'asynmétrique !
Le cryptographie asymétrique est un système qui se base sur un savoir en plus de la part de l'un des cotés, qui est une brèche permettant de déchiffrer facilement. La brèche, inconnue des attaquants, est ce qui garantie la sureté du système.
Vous décidez d'essayer d'implémenter le chiffrement RSA, qui est actuellement le chiffrement le plus utilisé au monde (banques, https..).
Dans le principe, c'est simple :
Si l'on as deux nombres premiers très grand et que l'on les multiplie, il sera extrèmement compliqué de trouver les deux nombres d'origine à partir du résultat.
Les seules méthodes sont de tester tout les nombres possible de 0 au nombre (avec un peu de reflexion, on peut simplifier cela à tout les nombres impairs jusqu'à la racine carrée du nombre). Ainsi, la force de l'algorithme repose sur la taille de la clé.
Note : Cela ne marche pas avec n'importe quel nombre, il faut impérativement

que les deux d'origine soient premiers.
Ainsi, le chiffrement asymétrique se fera en partageant le nombre obtenu à la multiplication des deux nombres premiers (clé publique). Il gardera les deux nombres secrets, pour pouvoir déchiffrer les messages (clé privée). Du fait qu'il est extrèmement complexe de trouver les deux nombres de base, la sécurité du système cryptographique se fera de cette façon. Les gens qui voudront communiquer n'auront qu'à chiffrer avec la clé publique (opération à sens unique) et feront parvenir le message au destinataire, qui déchiffrera avec sa clé privée.

Maintenant : Le fonctionnement
Il faut commencer par prendre deux nombres premiers, p et q. On calcule ensuite n = p * q et o = (p - 1) * (q - 1).
Ensuite, il faut choisir e un nombre tel que e n'ai aucun diviseur commun avec o.
Il faut ensuite trouver le nombre f compris entre 1 et o-1 tel que e * f - 1 soit un multiple de o.
Avec e et f, nous avons tout ce qu'il nous faut ! Nous nous servirons de n et e pour chiffrer, on peut donc les diffuser à tout le monde sans soucis. Pour déchiffrer cependant, nous utiliserons n et f.

Le chiffrement
Maintenant, comment chiffrer. Pour les exemples, nous utiliserons des nombres extrèmement petits. En pratique, il en faut des bien plus grands(superieur à 10$\hat{2}$0 10$\hat{3}$0, plusieurs dizaines de chiffres dans leur écriture décimale).
Par exemple, on as n = 851, e = 53 et f = 269.
Si l'on veut chiffrer par exemple les nombres 234 et 591, voici comment procéder :
La personne, n'ayant que la clé publique (n et e), doit calculer le nombre qu'elle souhaite chiffrer à la puissance e, et prendre le reste de la division de ce nombre par n (modulo).
Par exemple, pour 234 cela donne :
= (234 ** e) % n
= (234 ** 53) % 851
= 404
Ainsi, une fois chiffrée, 234 donne 404.
Ensuite, une fois le message chiffré (404 813), on envoie les nombres au destinataire qui va les déchiffrer avec sa clé privée.
Pour cela, il procède de la même façon que le chiffrement, mais avec f au lieu de e.
Cela donne :
= (404 ** f) % n
= (404 ** 269) % 851
= 234
Ainsi, il à déchiffré le message et retrouve les nombres d'origine (234 591).

Plus qu'à l'implémenter !

L'implémentation de tout cela doit aller, seulement, vous rencontrez des petits problemes.
Vous ne devez pas simplement faire passer des nombres, mais des messages. Pour passer du message aux nombres, c'est facile, vous n'avez que à prendre les valeurs ascii de ces dernières. Seulement, pour passer des nombres finaux à un message, vous ne voyez pas vraiment. En effet, par exemple s'il faut mettre 851 en ascii, vous ne pouvez pas le faire un un caractère, et s'il faut le faire en 2, vous n'avez aucun moyen de savoir que ce n'est pas deux nombres. Seulement, il vous vient tout à coup une idée en tête : réutiliser le bit validateur ! Au lieu de l'utiliser pour savoir si le message est fini ou non comme avec l'image (le message à une fin naturelle de sa taille, donc inutile), il va servir à voir si vous envoyez un nombre en plusieurs caractères ou non.
Pour par exemple, si l'on doit chiffrer les nombres 404, 38 et 813 (110010100, 100110 et 1100101101) cela donnerais :
1(bit validateur de nouveau nombre)00000001-0(bit validateur comme quoi le nombre continue)10010100-1(Nouveau nombre)00100110-1(Nouveau nombre)00000011-0(Nombre continue)00101101
Ainsi, une fois les explications enlevées :
100000001010010100100100110100000011000101101
Si besoin, l'on peut mettre des 0 au début pour en faire un multiple de 8 pour le transformer en ascii, le fait que ça commence par un 1 permet de bien gérer ça.
Cela donne 000100000001010010100100100110100000011000101101.
Une fois transformé en ascii, cela deviens :
<0x10><0x14>¤<0x9a><0x06>- (je n'ai pas pu écrire les caractères normalement, car ils sont assez particuliers)
Peu de caractères compréhensibles certe, mais c'est de l'ascii qui est possible de transmettre.
De plus, vous pouvez faire qu'au lieu de prendre seulement la valeur ascii pour chiffrer, ça prenne un "bloc" de m bits, m étant le plus grand nombre tel que 2**m <n.

Vous devez essayer d'implémenter le système RSA ! Bonne chance !