

Divers

D'après Mines 2015

1. On stocke des nombres sur 10 bits signés en complément à 2. Quels sont les nombres ainsi représentés ?
2. Un message est envoyé sous la forme d'une chaîne qui commence par une lettre indiquant le type de transmission ('A', 'B' ou 'C'), puis le nombre de données contenues dans le messages, sur 3 caractères, en décimal, ensuite les données : un caractère pour indiquer le signe, puis un nombre sur trois caractères, enfin, un checksum, qui correspond à la somme des données envoyées modulo 1000, sur 4 caractères (sans compter le nombre de données transmises).
 - (a) Compléter le message suivant : "A003+123-654+928"
 - (b) Écrire une fonction `lire` qui prend en paramètre une chaîne `c`, un nombre `n` et utilise un nombre `index` global pour lire les `n` caractères de `c` à partir de `index`, puis modifie `index` pour qu'un nouvel appel à cette fonction lise les caractères suivants.
 - (c) Écrire une fonction qui prend une paramètre un message sous forme de chaîne, et renvoie le triplet (type de transmission, liste des données, checksum). Le mode de transmission doit être correct.
 - (d) Écrire une fonction `affichage` qui prend en paramètre un message, et trace la courbe des données envoyées en fonction du temps. On considère que les données proviennent de mesures effectuées toutes les 2 ms.
 - (e) Écrire deux fonctions : l'une calculera la moyenne des données, l'autre son écart-type.

Codage de Huffman

Le codage de Huffman utilise un code à longueur variable pour représenter différents symboles. Il dépend de la fréquence attendue de ces symboles, et nécessite un prétravail sur un texte supposé représentatif.

1. Écrire une fonction qui prend une (longue) chaîne de caractères `ch` et renvoie un dictionnaire des nombres d'occurrence de chaque symbole. Par exemple, dans le dictionnaire associé à "dictionnaire", on trouvera la valeur 2 pour la lettre `n`.
2. Écrire une fonction qui prend un dictionnaire dont les valeurs sont des entiers et renvoie la somme des valeurs.
3. Écrire une fonction qui prend une liste d'objets pondérés (par exemple sous forme de couples), renvoie la liste actualisée dans laquelle les deux objets pondérés par les poids les plus faibles sont dans un couple, pondéré par la somme de leurs poids.
Par exemple, la liste `[(a',12),(b',18),(c',10),(d',4)]` devient la liste `[(a',12),(b',18),((c',10),(d',4),14)]` : les deux derniers objets, de poids minimum, ont été regroupés.
4. Écrire une fonction qui itère ce processus jusqu'à obtenir un seul élément.
5. La structure obtenue est arborescente. Les chemins dans l'arbre sont en bijection avec les caractères du texte. Pour aller plus loin, on peut envisager d'écrire des fonctions qui
 - crée à partir d'un tel arbre un dictionnaire associant à chaque caractère le chemin dans l'arbre (une suite de 0 et de 1, par exemple, 0 lorsqu'on descend vers le sous-arbre gauche, 1 vers le sous-arbre droit) pour parvenir à la feuille qui porte ce caractère

- encode un message en utilisant la fonction précédente
- décode un message en lisant dans l'arbre les 0 et des 1 comme des instructions pour se déplacer depuis la racine vers une feuille.

D.M.

1. Terminer ce T.P.
2. Chercher comment ouvrir un fichier **en lecture** et **en écriture**. Créer un petit jeu (demander à un joueur de deviner un nombre préchoisi aléatoirement ; morpion ; autre au choix) qui permet d'enregistrer des statistiques dans un fichier : nombres de parties jouées, nombre de parties gagnées, plus longue suite de parties gagnées, taille de la suite de parties gagnées en cours...
3. Commencer à réviser l'ensemble des T.P. et cours