

Structures de Données - L2, S4

TP 1 - Structures de bases et tableaux

Exercice 1.1 *Machine Primaire*

On dispose d'un agent ne sachant faire que des additions et multiplications de chiffres (et sait entre autres extraire le chiffre des dizaines). On se propose de modéliser l'agent et l'utiliser pour faire des opérations arithmétiques.

1. Proposer une structure `nombreA2Chiffres` avec deux champs qui permet de représenter des nombres à deux chiffres. (Attention on ne doit pas pouvoir stocker dans notre structure autre chose que deux chiffres.)
2. Écrire une fonction `arith` qui prend deux chiffres x et y , un booléen b et retourne $x * y$ (si $b = 1$) ou $x + y$ sinon sous la forme d'un objet de type `nombreA2Chiffres`. Votre fonction ne doit accepter que des chiffres en entrée (vous devrez gérer les erreurs possibles).
3. En utilisant la fonction `arith` écrire une fonction qui calcule la somme de deux nombres (les chiffres d'un nombre sont donnés sous forme d'un tableau).
4. Faites de même pour la multiplication de deux nombres.
5. Écrire un programme qui prend en entrée deux entiers et un caractère 'a' ou 'm' et fait une addition (si 'a'), ou une multiplication (si 'm') et retourne une erreur dans tous les autres cas. (Vous vous informerez sur comment on peut donner en C des arguments aux programmes comme sous Linux avec -)

Exercice 1.2 *Le problème de Josephus*

Lors de la révolte des juifs romains, un groupe de 40 juifs sachant la défaite imminente, décida de se tuer pour ne pas devenir esclaves. Pour cela, ils s'arrangèrent autour d'un cercle et en tournant autour du cercle (dans le sens des aiguilles) tuèrent toujours la 7ème personne vivante. Écrire un programme qui affiche l'ordre dans lequel les 40 personnes sont tuées.

Exercice 1.3 *Machine de Turing*

On se propose ici d'implémenter une machine de Turing. On rappelle ci-dessous ce qu'est une machine de Turing.

Une machine de Turing est une machine qui dispose d'un ruban divisé en cases et dans chaque case on peut écrire soit 1, soit 0, soit B (qui signifie case vide), soit le caractère $\#$. Au début d'un programme la machine commence toujours au début du ruban et utilise une variable (notée q) où elle stocke l'entier 2. La machine n'accepte qu'un seul type d'instructions :

lire c n n' c' d

- n, n' sont des entiers supérieurs ou égaux à 2,
- c, c' sont des caractères dans l'ensemble $\{0, 1, B, \#\}$,
- d est un caractère dans l'ensemble $\{l, r\}$.

La signification de l'instruction est la suivante : Lire la case courante du ruban, si c'est égal à c et que q est égal à n , alors affecter n' à q , écrire dans la case courante le caractère c' et aller à gauche (si $d=l$), sinon aller à droite. (On suppose le ruban infini vers la droite, ie on peut toujours aller vers la droite. Un programme dans cette machine s'exécute de la manière suivante : A chaque cycle d'horloge la machine regarde toutes les instructions et exécute la première possible, et si aucune des instructions ne peut être exécutée elle s'arrête et le résultat du programme est la séquence de bits sur le ruban.

1. Proposer une structure instruction qui permet de stocker une instruction de type lire c n n' c' d .
2. Un programme on va le voir comme la donnée d'un tableau d'instructions et un tableau (de taille MAX) qui représentera le ruban (dans notre cas il n'est pas infini).
Écrire une fonction qui prend un programme et qui l'exécute (vous gérerez les types d'erreur, eg, lorsque le ruban est plein et que l'on va aller vers la gauche ou la droite, ...).
3. Tester votre fonction avec par exemple les programmes écrits en TD (cette question peut attendre l'exercice 1.7).

Exercice 1.4 *Tri Fusion*

La fusion de 2 tableaux triés T_1 et T_2 consiste à produire un tableau T trié et contenant les éléments de T_1 et T_2 . Écrire une fonction réalisant la fusion. En utilisant la fusion, écrire une fonction qui trie un tableau sans utiliser de tableaux intermédiaires.

Exercice 1.5 *Chaîne de caractères*

1. Écrire un programme qui prend une chaîne de caractères et qui compte les espaces, les tabulations et les fins de ligne.
2. Écrire un programme qui prend une chaîne de caractères en entrée et qui le copie en remplaçant chaque série d'espaces par un seul espace.
3. Écrire un programme qui prend une chaîne de caractères en entrée et un caractère c et retourne soit le nombre de ligne ($c='l'$), ou le nombre de mots ($c='w'$) ou le nombre de caractères ($c='c'$).
4. Écrire un programme qui affiche un histogramme (vertical) des longueurs des mots qu'il reçoit en entrée.

Exercice 1.6 *Agenda*

Une page dans un agenda correspond aux sept jours de la semaine et pour chaque jour aux dix heures d'une journée de travail (8h-18h). Chaque heure on lui associe une chaîne de caractères (correspondant à un événement).

1. Proposer un type pour manipuler une page d'un agenda (attention on ne doit pas pouvoir enregistrer n'importe quoi dans notre agenda).
2. Écrire une fonction qui permet d'entrer un événement dans une page d'un agenda.
3. Écrire un programme qui permet d'écrire dans une page d'un agenda et d'afficher une page d'un agenda. A l'affichage, on doit avoir les heures en ligne et les jours en colonnes.

Exercice 1.7 *Fichiers*

En C pour manipuler les fichiers on utilise les fonctions `fopen`, `fprintf`, `fscanf`, `fclose` pour ouvrir, lire, écrire, fermer un flot sur un fichier (vous faites `man fprintf` pour avoir plus d'informations, de même pour les autres fonctions). Si vous programmez en Java vous lirez la documentation de Java pour savoir comment lire/écrire dans un fichier.

Écrire un programme qui prend en entrée un nom de fichier contenant un programme de la machine de Turing et qui l'exécute. Le résultat devra être stocké dans un deuxième fichier donné en argument. Le fichier contenant le programme est ainsi constitué :

- la première ligne contient les valeurs des cases du ruban (séparées par un `'`;`'`),
- chaque ligne restante est une instruction (on peut considérer que c'est de la forme `c;n;n';c';d`).