

05 jan 2015

Examen du module ALGO2

Durée : 1h30

Exercice 1 (9pts)

1. Soit un vecteur t d'entiers de taille $n \leq 50$. Ecrire une fonction **Plateau** qui vérifie si t est un plateau, c'est-à-dire si toutes les valeurs de t sont *égales*.

Exemple : $t = 5 \ 5 \ 5 \ 5 \ 5 \ 5$ est un plateau

2. On considère un autre vecteur d'entiers v de taille $m \leq 200$. En utilisant la fonction Plateau, écrire une fonction **PlusLongPlat** qui détermine le plus long plateau de v (c-à-d la plus longue séquence de valeurs consécutives *égales* dans v). La fonction doit renvoyer l'*indice de début* de la séquence ainsi que la *longueur* de la séquence.

Exemple : $v = 12 \ 5 \ 5 \ -1 \ 4 \ 2 \ 3 \ 3 \ 3 \ 3 \ 42 \ 5 \ 5 \ 5$

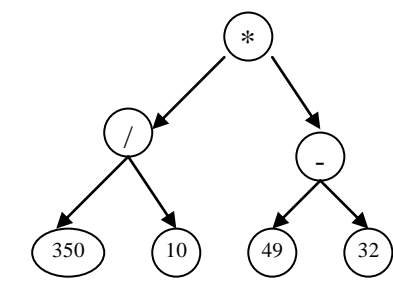
Le plus long plateau est la séquence **3 3 3 3**, son **indice de début = 6** et sa **longueur = 4**

3. Soit une matrice A ($n \leq 100$, $m \leq 200$) d'entiers. On voudrait récupérer la description (indice de début et longueur) du plus long plateau pour chaque ligne de la matrice A et sauvegarder ces deux informations avec le numéro de ligne) dans une structure de liste chaînée adéquate.
- Donner la déclaration de la liste.
 - Ecrire la fonction **ConstList** qui construit la liste en question, de telle manière que la liste soit triée par *ordre décroissant* des longueurs de plateaux (c-à-d la description du plus long plateau de la matrice doit se trouver en tête de liste)
4. Ecrire le programme qui remplit une matrice A ($n \leq 100$, $m \leq 200$) d'entiers, construit la liste chaînée des plus longs plateaux de A et affiche ces plus longs plateaux (il s'agit d'afficher le numéro de ligne et la séquence de valeurs constituant le plus long plateau de la ligne)

Exercice 2 (11pts)

Soit une expression arithmétique représentée par un arbre binaire où les nœuds sont des chaînes de caractères (de longueur ≤ 5) correspondant à un *opérateur* ou un *opérande*. Pour simplifier, on ne considère que les opérandes entiers et les opérateurs $+$, $-$, $*$, $/$ et $\%$.

Exemple



L'arbre correspond à l'expression
(350/10) * (49-32) qui vaut 595

1. Ecrire une fonction **Opérateur** qui accepte en entrée un caractère *c* et vérifie si c'est un opérateur (+, -, *, / , %) ou non.
2. Ecrire une fonction **Opération** qui accepte en entrée deux opérandes *op1* et *op2* de type *entier* et un opérateur *op* de type *caractère*, calcule et renvoie le résultat de l'opération (*op1 op op2*).
3. A partir de l'arbre contenant l'expression arithmétique, on voudrait récupérer la forme *postfixée* de l'expression dans une pile de chaînes de caractères.
 - Donner la définition du type *arbre* précédemment décrit.
 - Donner la définition du type pile de chaînes de caractères
 - Ecrire les primitives **Empiler** et **Dépiler**.
 - Ecrire une fonction **Postfixe** qui remplit la pile à partir de l'arbre, en effectuant un parcours postfixé.
4. Ecrire une fonction **Eval** qui évalue l'expression arithmétique à partir de la pile construite. On utilisera la fonction `int atoi (char *)` définie dans `<stdlib.h>` pour transformer une chaîne de caractères (numériques) en l'entier correspondant.

NB: si la fonction **Opérateur** (de la question 1) renvoie 0, il s'agit alors d'un opérande.
5. On suppose que l'arbre de l'expression arithmétique est déjà construit. Ecrire le programme qui construit la pile contenant la forme postfixée, évalue l'expression et affiche le résultat de l'évaluation.

Bon courage