

DS2 - Algorithmes et Structures de Données

Mardi 22 Janvier 2013

Durée 3H – Documents non autorisés

1. Récursivité et Pile - 4 pts

t: 6 3 0 9 1 7 8 2 5 4

Simulez la **pile** sur l'appel `trier-fusion(t,1,10)` (cf résumé de cours) en donnant aussi les valeurs intermédiaires de t après l'appel de `fusionner`.

2. Fichiers textes - 4 pts

Nous disposons d'un fichier texte contenant des notes (séparées par des espaces) : les notes sont disposées en colonne par matière et en ligne par élève. Nous souhaitons obtenir un nouveau fichier contenant les notes en colonne par élève et en ligne par matière.

Exemple :

10,5	18	15	16,5	8		10,5	12	8,5
12	11,5	10	14	17,5	→	18	11,5	9
8,5	9	11	7	6		15	10	11
						16,5	14	7
						8	17,5	6

Ecrivez un programme qui réalise cette transformation. Vous pouvez le découper en fonctions ou procédures, comme bon vous semble.

3. Evaluation d'une expression préfixée - 4 pts

3.1. Quelles sont les écritures préfixée et postfixée de l'expression infixée $(2*x+y)*(y-z)$?

3.2. On souhaite évaluer une expression préfixée. Nous rappelons qu'une expression préfixée est soit de la forme opérateur expression expression soit de la forme nombre.

Expliquez en français la méthode **récursive** de cette évaluation puis écrivez en pseudo langage la fonction **récursive** `evalPrefixe` qui réalise cette évaluation.

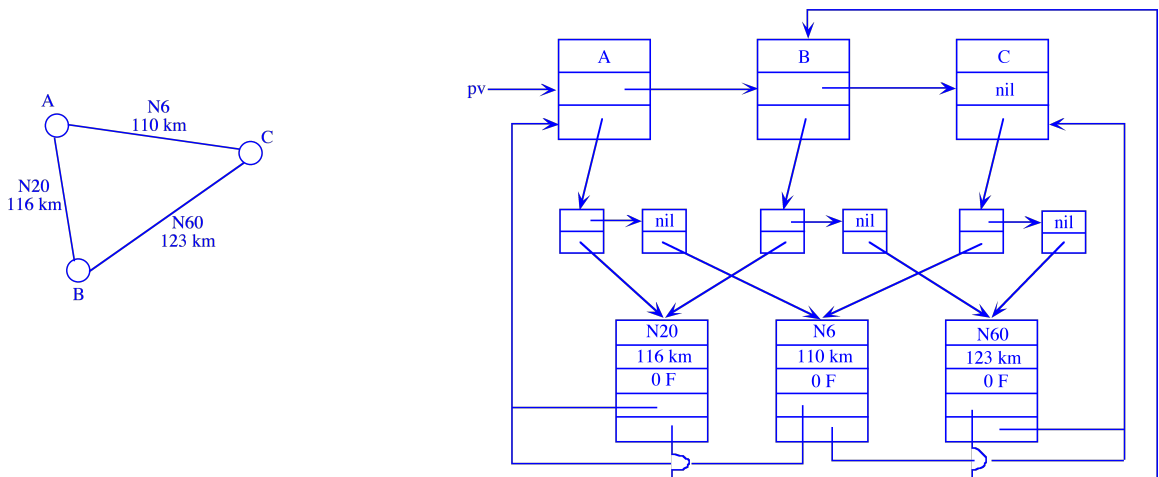
Fonction `evalPrefixe (c : chaine) : réel`

On pourra utiliser (entre autres) les fonctions `lire-mot(c,i)`, `effectuer1(m,x)`, `effectuer2(m,x,y)` et `traduire-C2R(m)` vues en TD (sur les piles).

4. Pointeurs - 8 pts

On représente le réseau routier d'un pays de la manière suivante : les villes sont des enregistrements chaînés les uns aux autres qui « pointent » également vers des enregistrements appelés « connexions routières » ; ces derniers sont chaînés entre eux lorsqu'ils concernent la même ville ; chaque connexion « pointe » de plus vers un « tronçon routier » qui indique le numéro de la route, la distance, le péage ainsi que l'origine et l'extrémité du tronçon (pointeurs sur des villes).

Exemple : la structure de données représentant les connexions entre les villes A, B et C donnent :



La description des structures de données utilisées est donc la suivante :

```

Type  listeVille = ^ville
      listeConRou = ^conRou
      listeTron = ^tron
      ville =      Enregistrement
                    nom : Chaîne
                    suiv : listeVille
                    connex : listeConRou

      conRou =      FinEnregistrement
                    Enregistrement
                    suiv : listeConRou
                    tron : listeTron

      tron =        FinEnregistrement
                    Enregistrement
                    route : Chaîne
                    dist, peage : entier
                    orig, extrem : listeVille
                    FinEnregistrement

```

4.1. On souhaite écrire la fonction `sontReliees` qui retourne un pointeur sur un tronçon routier entre les villes A et B (données par leur nom), si un tel tronçon existe. Dans le cas contraire, la fonction retourne « nil ».

Expliquez le principe en français puis écrivez en pseudo-langage cette fonction.

Fonction `sontReliees (A, B : chaîne) : listeTron`

4.2. On souhaite écrire une procédure `donneVille` qui retourne le nom des deux villes reliées par un tronçon routier donné par son nom (le tronçon existe).

Expliquez le principe en français puis écrivez en pseudo-langage cette procédure.

Procédure `donneVille (E nomtron : chaîne, S A, B : chaîne)`