ASI3 Mardi 23 Janvier 2007

Algorithmique et Bases de la programmation

Durée : *3h*
Documents autorisés : **AUCUN**

Remarques :
- Veuillez lire attentivement les questions avant de répondre.
- Le barème donné est un barème indicatif qui pourra évoluer lors de la correction.
- Rendez une copie propre.

1 Tri par fusion (2,5 points)

Donnez le corps de la procédure suivante qui permet de trier un tableau d'entiers à l'aide de l'algorithme du tri par fusion.

- **procédure** trierParFusion (**E/S** t : **Tableau**[1..MAX] **d'**Entier ; **E** debut,fin : **Naturel**)

Vous donnerez aussi les algorithmes de toutes les fonctions/procédures qui sont utilisées par cette dernière.

2 Position d'une sous-chaîne (2,5 points)

Donnez le corps de la fonction suivante qui retourne l'indice de la première occurrence de *sousChaine* dans *chaine*. Si *sousChaine* n'est pas présente dans *chaine*, alors la fonction retourne -1.

- **fonction** position (chaine,sousChaine : **Chaîne de caractères) : Entier**

Vous pouvez utiliser les fonctions suivantes :

- **fonction** longueur (uneChaine : **Chaîne de caractères) : Naturel**
  ...avec $longueur("") = 0$
- **fonction** iemeCaractere (uneChaine : **Chaîne de caractères**, iemePlace : **Naturel) : Caractère**

  ⌊**précondition(s)** $0 < iemePlace$
  $iemePlace \leq longueur(uneChaine)$

3 Un peu de C (3 points)

Soit le programme C suivant :

```c
#include <stdlib.h>
#include <stdio.h>
#define TAILLE_BUFFER 100

char* saisirChaine() {
  char buffer[TAILLE_BUFFER];
  scanf("%s",buffer);
  return buffer;
}

void afficherChaine(char* chaine) {
  printf("Vous avez tapé %s\n",chaine);
}

int main() {
  afficherChaine(saisirChaine());
  return EXIT_SUCCESS;
}
```

On compile ce programme :

```
> gcc -c -Wall -pedantic main.c
main.c: In function 'saisirChaine':
main.c:8: warning: function returns address of local variable
```

Répondez aux questions suivantes (à chaque fois en 10 lignes maximum) :

1. Que signifient les options `-c`, `-Wall` et `-pedantic` de *gcc* ?
2. Que signifie le *warning* affiché par *gcc* ? Expliquez comment résoudre ce problème (donnez le code C) ?

4 Distance dans un arbre binaire (8 points)

4.1 Question de cours (3 points)

4.1.1 TAD

Donnez le TAD ArbreBinaire et le TAD Dictionnaire à l'aide du formalisme vu en cours (sans la partie axiomatique et sémantique). Donnez aussi la signature des fonctions et procédures de leurs opérations.

4.1.2 Parcours

Soit l'arbre présenté par la figure n° 1. Donnez les parcours RGD, GRD et GDR.

4.2 Dictionnaire de pères (2,5 points)

Donnez le corps de la fonction `obtenirPères` suivante :

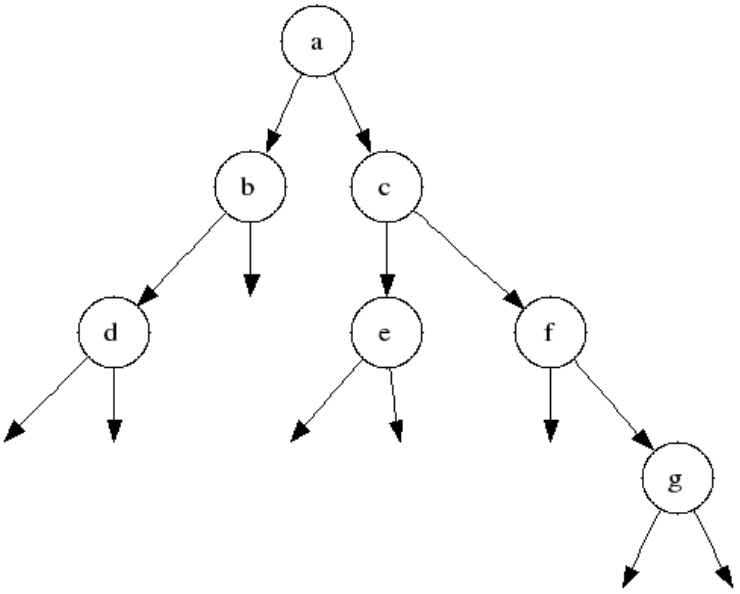

FIG. 1 – Un arbre de caractères

– **fonction** obtenirPères (a : ArbreBinaire<Element>) **:** Dictionnaire<Element,Element>

⌊**précondition(s)** non estVide(a)

...qui pour un arbre binaire non vide d'éléments distincts permet d'obtenir un dictionnaire dont :

– les clés sont les éléments de l'arbre qui possède un père (c'est-à-dire tous les éléments de l'arbre sauf celui contenu dans la racine)
– la valeur associée à chaque clé est l'élément qui se trouve dans le nœud père.

Par exemple, pour l'arbre présenté par la figure n° 1, le dictionnaire correspondant possédera les couples (clé,valeur) suivante : {('b','a'),('c','a'),('d','b'),('e','c'),('f','c'),('g','f')}

4.3 Distance dans un arbre (2,5 points)

Donnez le corps de la fonction suivante :

– **fonction** distanceDansUnArbre (a : ArbreBinaire<Element>, e1,e2 : Element) **: Naturel**

⌊**précondition(s)** non estVide(a)

...qui pour un arbre binaire donné non vide d'éléments distincts et deux éléments de cet arbre donne la distance en nombre d'arêtes qu'il y a entre les deux nœuds stockant ces deux éléments. Par exemple dans l'arbre présenté par la figure n° 1, la distance entre 'd' et 'c' est de 3.

5 Parcours d'un arbre binaire en largeur (4 points)

5.1 Hauteur d'un arbre (1 point)

Donnez le corps de la fonction suivante qui retourne la hauteur d'un arbre binaire (pour rappel la hauteur d'un arbre vide est de -1) :

- **fonction** hauteur (a : ArbreBinaire<Element>) **: Naturel**

5.2 Enfiler des éléments d'un certain niveau (1,5 points)

Donnez le corps de la procédure suivante qui enfile dans f tous les éléments d'un arbre a se trouvant au niveau n :

- **procédure** enfilerElements (**E/S** f : File<Element> , **E** a : ArbreBinaire<Element>, niveau : **Naturel**)

  ⌊**précondition(s)** $0 \leq$ n
  n $\leq$ hauteur(a)

5.3 Parcours en largeur (1,5 points)

Donnez le corps de la fonction suivante :

- **fonction** parcoursEnLargeur (a : ArbreBinaire<Element>) **:** File<Element>

...qui à partir d'un arbre a retourne une file qui contient tous les éléments de a rangés suivant un parcours en largeur.
Par exemple la file correspondante à l'arbre de la figure n° 1 est ('a','b','c','d','e','f','g') avec 'a' en tête de file.