

# CORRECTIONS DES EXERCICES DU CONCOURS D'ENTREE EPITA 2013

Pour toutes les fonctions demandées, plutôt que de fournir des solutions en C, Caml ou Pascal, nous avons préféré fournir une solution algorithmique en pseudo-langage facilement implémentable dans un de ces langages.

## A) Un peu de vocabulaire ...

- 1) Les noeuds internes (en ordre hiérarchique) de l'arbre B sont : **TWSCETU**
- 2) La taille de l'arbre B est : **13**
- 3) La longueur de cheminement externe de l'arbre B est : **18**
- 4) Le parcours *infixe* de l'arbre B est : **ICNWOEDTTESRU**

## B) Recherche ...

### Spécifications :

La fonction *cherche*(entier  $x$ , t\_vect11entiers  $V$ , entier  $d$ ) : entier retourne la position de la première occurrence de  $x$  trouvée à partir de la position  $d$  ( $1 \leq d \leq 11$ ) dans le vecteur  $V$ , la valeur 0 si  $x$  n'est pas présent.

```
algorithme fonction cherche : entier
parametres locaux
    entier          x          /* valeur cherchée */
    t_vect11entiers V
    entier          d          /* position de départ */
variables
    entier    i
début
    i <- d
    tant que (i <= 11) et (V[i] <> x) faire
        i <- i+1
    fin tant que
    retourne (i mod 12) /* modulo */
fin algorithme fonction cherche
```

## C) Cherche le 0 ...

### Spécifications :

La fonction *position0* (t\_vect11entiers  $V$ ) : entier retourne la position de la première valeur 0 non précédée de 9 dans le vecteur  $V$ . Elle retourne 0 si deux valeurs 0 se suivent et sont précédées de 9.

```
algorithme fonction position0 : entier
parametres locaux
    t_vect11entiers jeu
variables
    entier    pos
début
    pos <- cherche(0, jeu, 1)          /* on cherche le premier 0 */
```

```

si (pos = 1) ou (jeu[pos-1] <> 9) alors
  retourne pos
sinon
  pos <- cherche(0, jeu, pos+1)
  si jeu[pos-1] = 0 alors
    retourne 0
  sinon
    retourne pos
  fin si
fin si
fin algorithme fonction position0

```

## D) 1, 2, 3 ...

### *Spécifications :*

La fonction *verifie* (*t\_vect11entiers V*) : *booléen* retourne un booléen indiquant si les 9 entiers de 1 à 9 sont dans les premières cases dans l'ordre croissant suivies de deux valeurs 0.

```

algorithme fonction verifie : booléen
parametres locaux
  t_vect11entiers  V
variables
  entier  i
début
  si (V[10] <> 0) ou (V[11] <> 0) alors
    retourne faux
  sinon
    i <- 1
    tant que (i <= 9) et (V[i] = i) faire
      i <- i + 1
    fin tant que
    retourne (i = 10)
  fin si
fin algorithme fonction verifie

```

## E) Réussite ...

### Procédure déplace

#### *Spécifications :*

La procédure *deplace* prend en paramètre le tableau de jeu, ainsi que la position de la valeur nulle (0) à échanger. Elle échange ce 0 avec la valeur qui suit (dans l'ordre des entiers) celle qui le précède dans le tableau, 1 s'il est en première position.

```

algorithme procedure deplace
parametres globaux
  t_vect11entiers  jeu
parametres locaux
  entier  pos
variables
  entier  val
début
  si pos = 1 alors
    val <- 1
  sinon

```

```

    val <- jeu[pos-1] + 1
  fin si
  jeu[cherche(val, jeu, 1)] <- 0
  jeu[pos] <- val
fin algorithme procedure deplace

```

### Procédure réussite

#### *Spécifications :*

La fonction *reussite(t\_vect11entiers V)*: *booléen* tente de réussir la réussite : elle retourne un booléen indiquant la réussite ou l'échec.

```

algorithme fonction reussite : booléen
parametres locaux
  t_vect11entiers   jeu
variables
  entier   pos0
début
  pos0 <- position0(jeu)
  tant que pos0 <> 0 faire
    deplace(jeu, pos0)
    pos0 <- position0
  fin tant que
  retourne (verifie(jeu))
fin algorithme fonction reussite

```