

Algorithmes et Structures de Données

Mardi 17 Novembre 2020

Durée 1H30 – Cours et TD NON autorisés

1. Réussite - 12 pts

Soit un tableau de 11 cases rempli aléatoirement des entiers de 1 à 9 et de deux valeurs nulles (égales à 0). Le principe de la réussite est le suivant :

- Echanger une valeur nulle (symbolisant une place vide) avec le successeur de l'entier contenu dans la case précédant la valeur nulle. Si la première valeur du tableau est nulle, on l'échange alors avec la valeur 1.
- Recommencer ce processus jusqu'à ce
 - qu'on soit bloqué (les valeurs nulles sont toutes deux placées après le 9)
 - ou que les 9 valeurs soient placées dans les 9 premières cases du tableau.

Exemple

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

1.1. Ecrire en pseudo-langage une fonction `Position` qui renvoie l'indice de la première bonne case vide du tableau, 0 sinon.

Une case vide est « bonne » si elle n'est pas précédée d'un 9 ou d'un 0 précédé d'un 9.

1.2. Ecrire en pseudo-langage une procédure `Déplace` qui utilise `Position` pour échanger la case vide avec l'entier successeur de celui contenu dans la case précédant cette case vide.

1.3. Ecrire en pseudo-langage une fonction `Vérifie` qui renvoie le caractère 'e' si on a abouti à un échec, 'r' si on a réussi, 'c' si on doit continuer à jouer.

1.4. Ecrire en pseudo-langage le programme complet qui donne les tableaux intermédiaires jusqu'à l'échec ou la réussite.

1.5. Ecrire en pseudo-langage une procédure `Sauvegarde` qui sauvegarde un tableau dans un fichier texte (une seule ligne). On dispose de la fonction `int2car` qui fait la traduction d'un chiffre en un caractère.

1.6. Ecrire en pseudo-langage une procédure `Charge` qui charge un tableau sauvegardé dans un fichier texte (une seule ligne). On dispose de la fonction `car2int` qui fait la traduction d'un caractère en un chiffre.

2. Pile - 4 pts

```

Fonction f(x,y : entier) : entier
Var res : entier
Début
  Si (x=1) ou (y=x)
    Alors res ← 1
    Sinon res ← f(x,y-1) {@1}+ f(x-1,y-1) {@2}
  FinSi
retourner(res)
Fin
  
```

Simuler la pile pour l'appel `écrire(f(3,5)) {@0}` dans le programme principal.

3. Récursivité - 4 pts

3.1. Ecrire en pseudo-langage une **fonction itérative** `Supprime-it` qui supprime toutes les occurrences d'une lettre dans un mot (chaîne de caractères).

3.2. Ecrire en pseudo-langage une **fonction récursive** `Supprime-rec` qui supprime toutes les occurrences d'une lettre dans un mot (chaîne de caractères). **Expliquer le principe.**