UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence Informatique
Info0101 – 2020/2021

Travaux dirigés n° 3

Tableaux

Exercice 1 (Opérations de base)

Dans les questions qui suivent, on considère des tableaux de réels, préalablement alloués et éventuellement remplis. Écrivez les fonctions/procédures qui réalisent les traitements suivants :

1°) Remplissage d'un tableau avec des valeurs aléatoires comprises dans un intervalle passé en paramètre. On suppose l'existence d'une fonction `aleatoire() : réel` qui retourne un réel aléatoire dans l'intervalle [0 ; 1].

2°) Affichage d'un tableau.

3°) Recherche du nombre d'occurrences d'une valeur donnée dans un tableau.

4°) Recherche de la valeur maximum d'un tableau.

5°) Recherche du premier indice de la valeur maximum d'un tableau.

Exercice 2 (Tableaux - références et passage de paramètres)

1°) Donnez l'état de la mémoire à chaque instruction de l'algorithme suivant :

```
Algorithme manipulationTableaux
Déclarations
     Variables
          t, u, v : tableaux d'entiers
          i : entier
Début
{1}  t ← allouer(3)
{2}  t[0] ← 1 ; t[1] ← 3 ; t[2] ← 7
{3}  u ← t
{4}  u[1] ← 9
{5}  v ← allouer(3)
{6}  Pour i allant de 0 à 2 Faire
{6.1}     v[i] ← t[i]
     FinPour
Fin
```

À la fin de cet algorithme, déduisez la valeur des expressions suivantes : `u == t` et `v == t`.

2°) Étant donnée la procédure suivante :

```
Procédure multConst(t : tableau d'entiers, k : entier)
Déclarations
     Variables locales
          i : entier
Début
{1}  Pour i allant de 0 à taille(t)-1 Faire
{1.1}     t[i] ← t[i] * k
     FinPour
Fin
```

Donnez l'état de la mémoire au fur et à mesure de l'exécution de l'appel `multConst(v,3)` (avec `v` de l'algorithme précédent).

3°) Étant donnée la fonction suivante :

```
Fonction opp (t : tableau d'entiers) : tableau d'entiers
Déclarations
     Variables locales
          taille, i : entiers
          v : tableau d'entiers
Début
{1}  taille ← taille(t)
{2}  v ← allouer(taille)
{3}  Pour i allant de 0 à taille-1 Faire
{3.1}     v[i] ← -t[i]
     FinPour
{4}  retourner(v)
Fin
```

Donnez l'état de la mémoire de l'exécution de l'instruction : `u ← opp(v)` (avec `v` resté dans l'état précédent).

Exercice 3 (Manipulation des binaires)

1°) Écrivez une fonction/procédure *binaire* qui prend un nombre entier en paramètre et qui retourne la représentation binaire de cet entier sur 16 bits.

2°) Écrivez un algorithme qui demande de saisir un entier et qui affiche la représentation binaire de cet entier.

Exercice 4 ("Split")

On dispose de trois tableaux de n entiers : t, t_1 et t_2. On suppose que t est entièrement rempli, et que les deux autres tableaux sont alloués et non initialisés.

Écrivez un algorithme qui répartit dans t_1 et t_2 les entiers non nuls de t, les négatifs dans t_1 et les positifs dans t_2, puis qui complète les cases des deux tableaux avec des 0.

Exercice 5 (Triangle de Pascal)

1°) Écrivez une procédure qui affiche la ligne n du triangle de Pascal. Les lignes sont construites les unes après les autres, en les déduisant les unes des autres.
Vous utiliserez ici deux tableaux `a` et `b` :

- `a` contenant une ligne, calculez la suivante dans `b` ;
- recopiez dans `a` le contenu de `b`...

2°) Même question, en utilisant un seul tableau (à une dimension).

Exercice 6 (Recherche dans un tableau)

1°) Écrivez une fonction `rechercher` qui retourne l'indice de la première occurrence d'une valeur donnée dans un tableau.

2°) Écrivez une fonction `rechercherDernier` qui retourne l'indice de la dernière occurrence d'une valeur donnée dans un tableau.

3°) Écrivez une fonction `rechercherTous` qui retourne les indices de toutes les occurrences d'une valeur donnée dans un tableau.

4°) Écrivez un algorithme qui demande à l'utilisateur une valeur à rechercher et qui réalise les recherches précédentes. Donnez l'état de la mémoire avant et après chaque appel de fonction.