

Median AG43 Automne 2014

Mardi 21 Octobre 2014 - Durée 1h30 – Aucun document autorisé.

Les exercices sont à répartir sur deux copies : Ex. 1-2 sur copie 1, Ex. 3-4 sur copie 2.

1 TIRAGE DU LOTO

Ecrire une procédure nommée `LOTO7` qui effectue le tirage aléatoire de 7 nombres compris entre 1 et 49.

Cette procédure admettra comme unique argument le tableau d'entiers `tirage []` comprenant, après appel de la procédure, les 7 nombres qui auront été tirés au sort.

Vous disposez de la fonction `rand()` n'admettant aucun argument et retournant un entier aléatoire compris entre 0 et 32767, de la procédure `randomize()` qui initialise le générateur de nombres aléatoires au début de la séquence, ainsi que de l'opérateur `%` qui permet d'obtenir le reste de la division entière de 2 entiers.

La procédure utilisera un tableau interne comportant initialement 49 éléments (tableau que vous devrez générer de sorte à ce qu'il contienne dans l'ordre croissant les nombres de 1 à 49). Les tableaux sont indicés par rapport à 0.

Afin d'éviter de tirer 2 fois le même nombre, le nombre tiré au sort est supprimé du tableau (en décalant vers la gauche tous les éléments situés à sa droite) et la longueur est mise à jour.

En fait ce n'est pas le nombre en lui-même que vous tirez au sort mais plus exactement son indice dans le tableau. ne pas oublier de placer le nombre tiré dans le tableau `tirage []`.

EXEMPLE DE FONCTIONNEMENT DE LA PROCEDURE :

Le tableau `n []` (indicé par rapport à 0) de 49 éléments contient à un moment :

`{1, 2, 4, 5, 6, 7, 8, 9,10,...49}`.

Notez que le 3 n'y figure plus (ce qui signifie que son indice initial 2 a déjà été tiré et que 3 a été placé dans le tableau `tirage`).

Si un 7 est maintenant généré de manière aléatoire, le 9 sera à son tour placé dans le tableau `tirage` et le tableau `n []` deviendra :

`{1, 2, 4, 5, 6, 7, 8, 10,...49}`

2 TRI DU TIRAGE DU LOTO

Ecrire une procédure nommée `TRI_LOTO7` qui triera par ordre croissant le tableau de 7 entiers généré dans l'exercice précédent tout en laissant le dernier élément tiré au sort à sa place (fin du tableau, c'est le numéro complémentaire). Le tableau à trier sera passé en argument.

3 NOMBRE D'OCCURRENCES DES LETTRES DE L'ALPHABET

Ecrire une procédure nommée `alpha_occurrences` qui compte le nombre d'occurrences (nombre de fois où l'on rencontre) de chaque lettre de l'alphabet dans une chaîne de caractères qui sera passée en paramètre (de même que le tableau des occurrences).

On supposera que les chaînes sont délimitées par un caractère NUL.

Vous pourrez utiliser (ou non) une chaîne interne qui contiendra les 26 lettres de l'alphabet dans l'ordre.

Vous disposez de la procédure `strupr(chaine)` qui convertit une chaîne en majuscules et les caractères accentués en caractères non accentués (exemple : é, e, ê deviennent E).

Les tableaux et chaînes sont indicés par rapport à 0.

Exemple :

Si `chaine="test"`, le tableau des occurrences contiendra :

```
{0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0,0}
```

après appel de la procédure.

4 CENTRAGE D'UNE CHAINE DANS UN BUFFER

Ecrire la procédure `centrer` qui centre une chaîne de caractères dans un `buffer`. Le `buffer` fait plus de 32 caractères, mais seuls les 32 premiers caractères de la chaîne à centrer seront traités. Le `buffer` est destiné à être envoyé à un afficheur de 32 caractères, aussi, votre routine considérera que le `buffer` fait 32 octets. Lors du centrage, si la chaîne est inférieure à 31 caractères, le centrage sera effectué en plaçant des espaces en début du `buffer` et en copiant la chaîne à la suite. A la fin de la chaîne, on placera le caractère NUL. A l'appel de la procédure `centrer`, le `buffer` contient déjà la chaîne, mais placée au début. On dispose de la fonction `strlen(chaine)`

Un `buffer` est simplement un espace mémoire comme un autre.

Exemple pour un `buffer` de 16 caractères :

```
buffer = 'BONJOUR'          * le buffer contient :
                             * BONJOUR[NUL]????????
                             *
centrer(buffer)             * Appel de la fonction
                             *
                             * Apres appel:
                             * ____BONJOUR[NUL]????
```