TD n°2 de langage C : bitwises, tableaux, chaînes

Exercice 1 (acquisition)

Réalisez une fonction `int f0(int x)` qui retourne un Booléen ayant la valeur vrai si et seulement si les 4 bits de poids faibles de la représentation binaire de x sont tous à 0.

Réalisez une fonction `int f1(int x)` qui vérifie selon les mêmes modalités si les 4 bits de poids faibles de x sont tous à 1.

Exercice 2 (acquisition)

Réalisez une fonction `int n1(int x)` qui retourne le nombre de bits ayant la valeur 1 dans la représentation binaire de l'entier x.

Exercice 3 (acquisition)

Réalisez une fonction `int cmpt(int t1[], int t2[], int n)` qui compare les contenus des n premières cellules des tableaux t1 et t2 et retourne un Booléen valant vrai si et seulement si les contenu des ces n premières cellules sont identiques entre les deux tableaux.

Exercice 4 (acquisition)

Réalisez une fonction `void ins(int t[], int i, int x, int n)` qui insère la valeur x en position i dans le tableau t, supposé avoir une longueur n. Les éléments initialement situés aux positions i à n-2 sont décalés. L'élément initialement situé en position n-1 est supprimé.

Utilisez un dessin pour expliquer le principe de votre solution.

Exercice 5 (acquisition)

Réalisez votre propre implantation de la fonction `void scat(char* dest, char* src)` ayant un comportement identique à celui de la fonction standard **strcat**, à savoir que la chaîne **src** est concaténée à la chaîne **dest**.

Exercice 6 (consolidation)

Réalisez une fonction void `exgb(int* p, int a, int b)` qui échange les valeurs des bits situés en positions a et b dans l'entier pointé par p.

Les position 0 correspond au bit de poids faible.

Exercice 7 (consolidation)

Réalisez une fonction void `iclass(int[] t1, int n1, int[] t2, int n2, int[] t, int n)` qui, en supposant que t1 contient n1 entiers classés en ordre croissant et que t2 contient n2 entiers classés en ordre croissant, place dans t tous les éléments de t1 et tous ceux de t2, classés en ordre croissant.

Exercice 8 (consolidation)

Réalisez une fonction `void gsc(int n, int k)` qui affiche toutes les suites strictement croissantes de n entiers de valeurs comprises entre 1 et k.

N'hésitez pas à décomposer votre solution en plusieurs fonctions et à utiliser la récursivité.

Exercice 9 (consolidation)

Réalisez une fonction `int t1c(char *s, int k)` qui s'exécute *en temps linéaire* et retourne un Booléen valant vrai si et seulement si aucun caractère n'apparaît plus de k fois fois dans la chaîne s.

Vous pouvez utiliser un tableau d'entiers.