TP 3 : Expressions complexes et traitement de bits

Programmation en C (LC4)

Semaine du 20 février 2012

1 Expressions complexes et effets de bord

Question 1 Définire une *macro* (pour le pré-processeur) `MAX(a,b)` qui sera remplacée avant la compilation par une expression s'évaluant en le maximum des valeurs de `a` et `b`. Faire de même une macro `ABS(a)` donnant la valeur absolue.

Question 2 Donner la définition d'un type de structure contenant un caractère, et un entier indiquant combien de fois ce caractère a été utilisé. Définir une macro `ACCES(s)` permettant d'accéder au caractère contenu dans une telle structure, et qui incrémente le compteur d'accès (cette macro doit pouvoir être utilisée dans n'importe quelle expression).

Question 3 Ecrivez une fonction `int fact(int x)` déclarant une variable `z` puis calculant en une seule instruction (et sans bloc {...}) la factorielle de `x`, dont la valeur est stockée dans `z` et enfin retournée.

2 Opérateurs de traitement de bits

Question 4 Ecrire une fonction `void affiche_binaire(int x)` qui affiche l'écriture binaire d'un entier donné `x`.

Question 5 Ecrire une fonction `int bit_parite(int x)` qui prend en argument un entier `x` de quatre octets, de valeur strictement inférieure à 2^{31}, et renvoie cet entier dans lequel le bit de poids fort (le bit plus à gauche) est un *bit de parité* : il vaut 0 si le nombre de 1 dans la représentation binaire de cet entier est impaire, et 1 sinon.

Question 6 Ecrire une fonction `int miroir(int x)` qui inverse les bits de l'entier `x`, pour que son i^{eme} bit devienne son $n - i^{eme}$ bit, s'il est représenté sur n bits.

3 Cryptage simpliste

On teste notre méthode naïve de cryptage, où l'on effectue une rotation des bits de chaque caractère du message pour modifier la séquence d'entier représentant le texte (cf. feuille de TD).

Question 7 Ecrire les fonctions `unsigned char lire_bits_g(unsigned char x,int n)` et `unsigned char lire_bits_d(unsigned char x,int n)` renvoyant le caractère contenant (à la même position) les n bits les plus à gauche (respectivement à droite) de `x`, et `0` aux autres positions.

Question 8 Ecrire les fonctions `unsigned char ecrire_bits_g(unsigned char x,unsigned char y,int n)` et `unsigned char ecrire_bits_d(unsigned char x,unsigned char y, int n)` renvoyant `x`, dans lequel les n bits les plus à gauche (respectivement à droite) ont été remplacés par les n bits les plus à gauche (droite) dans `y`.

Question 9 Ecrire les fonctions `unsigned char rotation_g(char x,int n)` et `unsigned char rotation_d(char x,int n)` réalisant la rotation de n position des bits de l'entier passé en arguement, à gauche et à droite respectivement.

Le cryptage se fait en appliquant une rotation de n positions (pour un certain n choisi, c'est donc notre *clé*) à chacun des caractères du texte à rendre secret.

Question 10 Ecrire un programme utilisant les fonctions précédentes pour crypter un texte fourni en entrée (ou par un fichier), et le programme permettant de le décrypter.

On a remarqué que cette méthode était excessivement naïve, puisque la clé est réduite à un entier entre 1 et 7. On veut donc pouvoir utiliser une clé plus grande, pour que le travail de décryptage soit plus difficile sans posséder la clé (si on doit la deviner pour casser le cryptage).

Question 11 Améliorer le programme en utilisant un tableau d'entiers comme clé (où l'on utilise chacun de ces entiers comme clé une fois, avant de réutiliser les mêmes valeurs). Imaginer d'autres amélioration pour rendre le message plus difficile à obtenir sans la clé.

4 Cryptage de Vernam

On souhaite implémenter une méthode de cryptage inventée par Gilbert Vernam, reposant sur l'utilisation de l'opérateur «ou-exclusif» entre un texte et une clé. Le principe est simple : on crypte le n^{eme} caractère du message en produisant le résultat d'un ou-exclusif bit-à-bit entre ce caractère et le n^{eme} caractère de la clé. On peut alors transmettre le message codé, et trouver un moyen sûr de transmettre la clé permettant de le décrypter.

Question 12 Ecrire une fonction `char coder(char x,char k)` qui encode le caractère `x` donné à l'aide du morceau de clé `k`.

Question 13 Ecrire un programme qui lit tout d'abord une clé sur l'entrée standard, puis lit un texte qu'il va crypter à l'aide de cette clé et afficher sur la sortie standard.

Cette méthode de cryptage est complètement sûre, elle garantit que le message crypté ne peut être déchiffré sans la clé. Son désavantage est qu'elle nécéssite normalement une clé de même taille que le message à transmettre (et que cette clé doit être transmise de manière sûre, typiquement sur un support physique et non par le réseau).

Question 14 Améliorer le programme en lui permettant de récupérer ou de générer un morceau de clé supplémentaire pour crypter un texte, si le texte est plus long que la clé initialement fournie (soit par génération automatique, soit par interaction avec l'environnement).

Le décryptage d'un message est très simple : on applique de nouveau la même méthode du ou-exclusif entre le message crypté et la clé, et on obtient le texte original (voir la table de vérité du ou-exclusif pour comprendre pourquoi). On peut donc réutiliser le même programme pour déchiffrer un message, en donnant simplement la même clé que lors du chiffrement.

5 Champs de bits et sondages

On va représenter le résultat d'un sondage concernant une éléction à l'aide de champs de bits. Le but est de savoir, pour chaque personne interrogée, quels sont les candidats (parmi 5) pour qui elle pourrait envisager de voter, ou pour qui elle refuserait de voter.

Question 15 Définir un type de structure indiquant les préférences de vote d'une personne, départageant parmi les candidats ceux qui sont acceptés ou rejetés (en le minimum de place). Une personne peut soutenir plusieurs candidats (elle n'est peut-être pas encore décidée).

Question 16 Ecrire un programme qui, pour un tableau représentant n personnes interrogées dans le sondage, calcule quel candidat est le mieux accepté (pour qui un maximum de gens seraient prêts à voter) et quel candidat est le plus rejeté (ayant un minimum de soutien). Comment affiner ce sondage, et notre structure de donnée ? Ce sondage nous donne-t-il le résultat de l'élection ?