CHALLENGE ALGORITHMIQUE GEII 2011

Le challenge d'algorithmique GEII 2011 consiste à retrouver un message caché dans un fichier.
Il est destiné au étudiants de DUT GEII Toulon, il n'y a rien d'autre à gagner qu'un gros paquets de bonbons. Le vainqueur est celui qui enverra le premier (l'heure du mail faisant foi) à arlotto@univ-tln.fr , le texte caché ainsi que le fichier source du programme utilisé pour le trouver et son mode d'emploi. Il n'y pas de date limite. Attention n'envoyez pas de fichier exécutable, votre message serait bloqué par le serveur de mail.
Le message est caché dans un fichier image au format bmp :

challenge.bmb à http://arlotto.univ-tln.fr/cours_de_c/challenge.

Il s'agit d'un texte en français d'environ 1500 caractères. Pour cacher le message sans altérer l'image, quelques bits de poids faibles des octets composant les pixels de l'image sont modifiés selon la méthode suivante :

-On choisit un octet de départ dans l'image et on modifie (éventuellement) son bit de poids faible de manière à ce qu'il soit égal au bit de poids faible du premier octet du message. Ensuite on se décale d'un certain nombre d'octets dans l'image (Ce nombre est appelé dilution) puis on modifie (éventuellement) le bit de poids faible de l'image pour qu'il soit égal au deuxième bit du premier octet du message. On se décale à nouveau de la dilution et on recommence la modification pour le prochain bit du premier octet du message. Après huit itérations, on a caché un octet. On répète alors le même procédé pour chaque octet du message en se décalant de la dilution entre chaque octet.
Le point de départ et la valeur de la dilution constitue la clé (le mot de passe) de ce procédé de stéganographie rudimentaire. La dilution est ici inférieure à 30 et le départ inférieur à 10000.
Exemple pour un dilution de 3 et un message composé de "AB" avec un départ de 1
"AB"="\x41\x42"=[01000001 01000010]

Octet dans l'image originale :

F4 **B6** 32 F2 **F5** E4 34 **00** 07 EE **B4** C6 F5 **F4** F4 45 **21** 34 45 **56** EF FB **AA** 23 21 **B4**
B7 34 **02** 15 13 **14** AA A1 **A3** A4 A5 **34** 45 F3 **ED** D5 23 **41** 01 02 **02** 03 02 **EF** EF BE

On démarre de la position 1 et seul un octet sur 3 est susceptible d'être modifié (en gras).
Octets dans l'image modifiée :

F4 **B7** 32 F2 **F4** E4 34 **00** 07 EE **B4** C6 F5 **F4** F4 45 **20** 34 45 **57** EF FB **AA** 23 21 **B4**
1 **0** **0** **0** **0** **0** **1** **0** **0**

B7 34 **03** 15 13 **14** AA A1 **A2** A4 A5 **34** 45 F3 **ED** D5 23 **41** 01 02 **02** 03 02 **EF** EF BE
1 **0** **0** **0** **0** **1** **0** **…...**

L'image originale est très peu altérée car très peu de bit sont modifiés et les bits modifiés sont toujours de bits de poids faible donc ils modifient très peu le rendu visuel de l'image.
Bien sûr, ce procédé très simple ne résiste pas à la compression de l'image ni au changement de format.
Pour manipuler facilement les bits en C, vous pouvez utiliser les macros du fichier : macro_bits.c
Attention, si vous utilisez windows il est important d'ouvrir les fichiers en mode "binaire" "rb".
L'utilisation du langage C n'est toutefois pas obligatoire.

Pour gagner il faut être le premier à retrouver au moins 90% du message original. Votre méthode n'a pas à être entièrement automatique (un petit nombre de vérifications manuelles est par exemple autorisé) mais il faut fournir un mode d'emploi précis permettant de la reproduire. Il est bien sûr interdit de poster le challenge sur des forums mais vous pouvez vous aider d'internet et/ou vous y mettre à plusieurs.
Avec un peu d'astuce, un bon algorithme et un brin de culture, votre programme devrait retrouver le texte caché en moins de 20 minutes sur un pc de bureau.
Bon courage.