



Examen de Programmation en C - IR1/OC 2009-2010

Polycopiés de cours et notes de TD autorisés Durée: 2 heures

Pour chaque fonction que vous écrirez, vous devrez définir soigneusement le prototype en tenant compte de tous les cas d'erreurs possibles, et vous mettrez tous les commentaires nécessaires pour que l'on sache comment il faut l'utiliser.

Exercice 1: Anna-Lise de Caux-Dessé (3 points)

Si, vous aussi, vous voulez devenir prof de C et passer derrière les élèves en TD en prédisant leurs erreurs avant gcc sous leur regard émerveillé, montrez votre valeur en faisant tous les commentaires pertinents sur la fonction suivante, censée calculer la valeur maximum d'un tableau:

```
int get_maximum(int[] t, int n) {  
    int n=t[0];  
    for (i=0;i<n:i++) {  
        if (t[i]>n) n=t[i];  
    }  
    return n;  
}
```

Exercice 2: pêche à la ligne (8 points)

Comme tous les pêcheurs qui se vantent sur leurs prises, vous voulez savoir qui a la plus grande. Puisque vous voulez pêcher des lignes, vous écrirez tout d'abord une fonction `peche_ligne` qui prend en paramètre un descripteur de fichier et qui lit une ligne complète (jusqu'à un saut de ligne ou la fin de fichier). Vous utiliserez de l'allocation dynamique pour être certain qu'il n'y ait pas de débordement de tableau. La fonction retournera un pointeur sur la chaîne contenant la ligne lue, ou `NULL` en cas de fin de fichier. Si la ligne se termine par un `\n`, celui-ci ne sera pas recopié dans la chaîne résultat. **(5 points)**

Vous devrez ensuite écrire un programme prenant en argument un nom de fichier et devant afficher sur la sortie standard la plus longue ligne de ce fichier. Pour cela, vous réutiliserez la fonction précédente et vous avez droit à `strlen`. Attention à ne pas laisser de fuite de mémoire dans votre code. **(3 points)**

Exercice 3: two bits or not two bits (3 points)

Bien qu'il ait arrêté ses études mathématiques en terminale pour se consacrer à une étude comparative sur les boissons anisées à travers le monde, votre beau-frère Gérard a trouvé dans un apéritif maya une idée géniale qui va révolutionner la cryptographie, pour peu que vous lui apportiez votre concours généreux. Tout ce qu'il faut à votre parent d'exception est une fonction prenant en paramètres 2 octets non signés et qui les compare bit à bit. La fonction renvoie le nombre de bits qui diffèrent. Par exemple, si l'on appelle la fonction avec les valeurs A et B suivantes, on doit obtenir ceci:

A=00110011
B=01011010
résultat=4 différences

Ayez donc la bonté de fournir cette fonction sur votre copie, le prof la transmettra à ce sacré Gégé au bistrot.

Exercice 4: Geolocalized Petanque System (7 points)

Félicitations, votre entreprise vient de décrocher le contrat de l'amicale des boulistes de Ladal-en-Pante pour un système d'arbitrage de pétanque par GPS. On rappelle que la pétanque se joue avec 2 équipes, disposant chacune de N boules. Il y a également un cochonnet. À la fin d'une partie, l'équipe gagnante est celle qui possède la boule la plus proche du cochonnet. Si la seconde boule la plus proche appartient à la même équipe, elle marque 2 points. Si la boule suivante également, elle marque 3 points et ainsi de suite. Pour simplifier, on considérera que les coordonnées sont en 2 dimensions et on négligera le diamètre des boules et du cochonnet, qui seront donc considérés comme de simples points.

Proposez une structure de données permettant de représenter l'état d'une partie terminée. Vous pouvez naturellement utiliser des sous-structures de données si nécessaire, mais vous devrez justifier et commenter vos choix. **(2 points)**

Sachant que la distance entre 2 points A et B est obtenue par la formule $\text{racine}((Xa-Xb)^2+(Ya-Yb)^2)$, écrivez une fonction prenant en paramètre une partie et retournant le vainqueur. Rappel: la racine carrée s'obtient avec la fonction `C sqrt`. **(3 points)**

Donnez une nouvelle version de la fonction précédente, pour qu'elle indique en plus combien de points sont attribués au vainqueur. **(2 point)**