DEV 1.1 : Examen Machine

16 Janvier 2025

1 Instructions

- Le contrôle dure 2h30.
- Vous avez droit à vos notes de cours, ainsi que les sujets de TPs à l'adresse http://www.iut-fbleau.fr/sitebp/dev11/ et le code de vos précédents TPs.
- Tout le reste, comme notamment les sites web autres, est proscrit.
- Le contrôle doit être fait sur une machine de l'IUT, sous OS Unix.
- Tout travail doit être exécuté de façon individuel.
- Chaque exercice sera dans son propre fichier avec l'extension .c .
- Vos réponses doivent être envoyées à luc.dartois@u-pec.fr avec le sujet : [ExamenDEV1.1] votre_Prenom votre_Nom
- Vous y joindrez une unique archive s'appelant *Prenom_Nom.tar* contenant les codes sources de vos exercices. Je rappelle la commande (à exécuter dans le dossier correspondant au contrôle) :

  ```
  tar cvf Prenom_Nom.tar *.c
  ```

- Les exercices peuvent être faits dans n'importe quel ordre.
- Vous pouvez créer des structures ou des fonctions non demandées si vous le jugez nécessaire.
- Lisez bien les questions. Ne perdez pas bêtement des points.
- Le non-respect des consignes exactes entraîne une perte de points pouvant aller jusqu'au 0.

2 Exercices

Exercice 1. Échauffement. (2 points)

Écrire un programme affichant tous les multiples de 13 compris entre 2025 et 20025.

Exercice 2. Cible. (4 points)

Écrire un programme qui demande un l'utilisateur un entier, puis dessine sur la sortie standard une cible de la taille de l'entier donné, telle que dans l'exemple ci-dessous.

```
Quelle taille pour votre cible ?
3
   +
   +
   +
+++++++
   +
   +
   +
```

Exercice 3. PGCD. (4 Points)

Le *plus grand diviseur commun* (pgcd) de deux entiers a et b est le plus grand nombre qui divise à la fois a et b. On le note $pgcd(a, b)$. On remarque que le $pgcd(a, 0) = a$ pour tout a, que $pgcd(a, b) = pgcd(a - b, b)$ si $a >= b$ et que $pgcd(a, b) = pgcd(b, a)$ si $a < b$.

1) Écrire une fonction récursive implémentant **cet algorithme**.

2) Implémenter **ce même algorithme** dans une fonction non récursive.

Exercice 4. Correcteur. (6 points)

On souhaite écrire un programme correcteur syntaxique.

1) Tout d'abord, écrire un programme lisant le fichier texte donné en premier argument, et le recopiant dans un nouveau fichier dont le nom devra être le même que le fichier original auquel on ajoute *Corrige*. On pourra s'appuyer sur les fonctions de la bibliothèque *string.h*.

2) Modifiez votre programme pour corriger les erreurs suivantes (traitez chaque type d'erreur comme une question séparée) :

- Les points doivent être suivis d'exactement un espace ou un retour à la ligne.
- Les phrases doivent commencer par une majuscule.
- Il n'y a pas de majuscule en dehors du cas précédent.

Vous pouvez si vous le souhaitez tester votre programme sur le fichier texte exemple.txt.

Exercice 5. Mélange. (6 Points)

1) Écrire une structure de liste chaînée dont les maillons sont des entiers.

2) Créer des fonctions permettant d'afficher une liste, ainsi que d'ajouter des éléments au début d'une liste.

3) Créez dans la fonction *main* deux listes chaînées, l'une contiendra dans l'ordre les entiers multiples de 3 et de 7 inférieurs à 50, et l'autre les multiples de 5 et de 13 inférieurs à 50.

4) Écrire une fonction prenant deux listes triées en entrées et renvoyant une nouvelle liste contenant la combinaison triée des deux listes, et testez la en affichant la combinaison des deux listes précédemment créées.