

Examen C – Session 1

ING1 – Programmation C

17 janvier 2013



Modalités

- Durée : 3 heures
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document papier n'est autorisé.
- Aucune sortie n'est autorisée avant une durée incompressible de 30 minutes.
- Aucun déplacement n'est autorisé.
- **Aucune question au professeur n'est autorisée.** Si le sujet ne vous semble pas clair, à vous d'expliquer dans des commentaires pourquoi est ce que ça ne vous semble pas clair, et quelles modifications vous effectuez pour réaliser l'exercice.
- Aucun échange, de quelle que nature que ce soit, n'est possible.
- Les différentes fonctions/procédures demandées seront toutes à tester dans un main.
- Vous traiterez chaque exercice dans un dossier différent.
- Le barème est indicatif, il intègre pour chaque question un test dans un main.
- La présence d'un Makefile et des commentaires doxygens sont un plus, et seront appréciés.
- La lisibilité du code et les commentaires seront pris en compte dans la notation, de même que la séparation des différentes procédures/fonctions dans des fichiers d'en-tête (.c .h).
- **Tout code qui ne compile pas ne sera pas corrigé et entraînera une note nulle. Si une partie de votre code ne compile pas, commentez-le. Ceci implique évidemment que vous compiliez au fur et à mesure!**
- Chaque warning de compilation entrainera une **pénalité de 0.5 point**.
- Tout ce qui sera dans le dossier **rendu** correspondra à votre rendu, et sera récupéré à la fin de l'examen. Je vous conseille vivement de travailler directement dans ce dossier.

1 Fractions entières (0.5 + 1 + 1 + 2 + 4 = 8.5)

- 1 Définir la structure `Fraction` qui est composée d'un numérateur et d'un dénominateur entiers.

 - 2 Écrire la fonction `saisirFraction` qui demande la saisie du numérateur et dénominateur à l'utilisateur et qui renvoie une variable de type `Fraction`.

 - 3 Écrire la procédure `afficherFraction` qui affiche une fraction sous la forme n/d . Si le numérateur est plus grand que le dénominateur (en valeur absolue), on affichera la fraction sous la forme $a + n1/d$ où a est un entier et $n/d = a + n1/d$.

 - 4 Écrire la méthode `formeIrreductible` qui permet de transformer une fraction en forme irréductible.
Une fraction irréductible est une fraction a/b où le numérateur a est un entier et le dénominateur b un entier positif, de telle façon qu'il n'existe pas d'autre fraction c/d représentant le même nombre avec c plus petit que a en valeur absolue, c et d étant également deux entiers.
Si une fraction n/d n'est pas irréductible, alors elle peut être réduite, c'est-à-dire que n et d peuvent être divisés par un même nombre, le plus grand possible. Ce nombre est le PGCD de n et d .
- Rappel :** $\text{PGCD}(a, b) = \text{PGCD}(a - b, b)$ ou $\text{PGCD}(a, b) = \text{PGCD}(b, r)$ avec r le reste de la division entière de a par b .
- 5 Écrire les différentes opérations binaires (addition, soustraction, multiplication et division) pour les fractions. Elles prendront en paramètre 2 fractions et retourneront le résultat sous forme de fraction.

2 Fleuriste (0.5 + 1 + 1 + 1 + 2 + 3 = 8.5)

Un fleuriste propose les fleurs suivantes : Rose (0.90€), Lys (1.10€), Germini (0.80€), Oeillet (0.70€), Alstroemeria (1€), Tulipe (0.60€) et Glaieul(1.20€). On supposera que chacune des fleurs est disponible dans les couleurs suivantes : rouge, blanc, rosée, jaune, orange, violet, bleu. Le prix

de chacune des fleurs augmente de 0.01€ en fonction la couleur (d'après l'ordre donné précédemment). Par exemple, une rose rouge coûte 0.90€, une rose blanche coûte 0.91€, une rose bleue coûte 0.96€....

On souhaite mettre en place un programme qui permet de gérer une commande et d'éditer une facture.

- 6 | Définir les énumérations couleur et fleur.

- 7 | Écrire les fonctions saisirCouleur et saisirFleur qui demandent à l'utilisateur la couleur ou la fleur voulue, et qui retourne la valeur associée.

- 8 | Écrire les procédures afficherCouleur et afficherFleur qui prennent en paramètre respectivement une couleur et une fleur et qui affichent la valeur associée.

La commande sera stockée dans un tableau static à deux dimensions d'entiers positifs : une première dimension correspondant aux fleurs, une deuxième correspondant aux couleurs et la valeur de la matrice correspondant à la quantité de fleurs commandée.

- 9 | Écrire la procédure afficherCommande qui prend en paramètre la matrice de commande et qui affiche l'état courant de la commande sous la forme Fleur Couleur : Quantité lorsque la quantité est non nulle. Par exemple :

Rose Rouge : 10
Germini Orange : 15

- 10 | Écrire la procédure saisirCommande qui prend en paramètre la matrice de commande et qui, en fonction du choix de fleur et de couleur de l'utilisateur, incrémente d'une quantité définie par l'utilisateur la case associée.
Tant que l'utilisateur n'a pas fini sa commande, la saisie continue. Pour un meilleur suivi, un affichage de l'état courant de la commande sera apprécié.

11

Écrire la procédure `enregistrerFacture` qui enregistre la commande dans le fichier “facture.txt”. La facture doit prendre en compte le détail du montant (prix unitaire, quantité). Par exemple :

```
-----
                        FACTURE
-----

Rose Rouge : 0.90 x 10 = 9.00
Germini Orange : 0.84 x 15 = 12.60

                        Montant total : 21.60
-----
```

□

3 Tableau dynamique (3)

Pour la fonction `main` de cet exercice, la taille du tableau dynamique sera défini par l'utilisateur, puis il sera initialisé aléatoirement¹, et la valeur du décalage sera saisie par l'utilisateur.

12

Écrire la procédure `decalage` qui prend en paramètre un tableau dynamique, sa taille et un entier correspondant au pas de décalage, et qui modifie le tableau de façon à décaler toutes les valeurs vers la gauche en fonction du pas.

Par exemple : `[3; 18; 2; 12; 7; 8]` décalé de 1 devient `[18; 2; 12; 7; 8; 3]` et décalé de 3 devient `[12; 7; 8; 3; 18; 2]`. □

1. Rappel : la fonction `rand` permet de générer un nombre aléatoirement. Pour plus d'information : `man 3 rand`.