



Rédigé par : Elizabeth R., Vincent G. et Rachid C.

Ref : *ING1-PROGC-EXAMEN*

A l'intention de : Etudiants des ING1

Créé le : 10/01/201

Fanteistic

Préambule

Cet examen dure 3h00. Le rendu de l'examen est une archive électronique (.zip ou .tar.gz) qui se rendra sur clef USB fournie par les examinateurs. Pendant cette épreuve, seul votre pc en local est autorisé (pas de documents papiers).

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 compilez au fur et à mesure!

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).

Vous traiterez chaque exercice dans un fichier source (.c) différent, les différentes fonctions/procédures demandées seront toutes à tester dans un *main*.

Le barème est indicatif, il intègre pour chaque question un test dans un *main*.

1. Polynômes du second ordre (0.5 + 1 + 1 + 2 + 2 = 6.5)

Question 1) Définir la structure *Monome* qui est composée d'un degré et d'un coefficient.

Question 2) On définit un polynôme du second ordre comme un tableau de **trois** *Monome*. Écrire la fonction *saisirPolynome* qui demande à l'utilisateur les valeurs des coefficients pour chaque *Monome* du polynôme, et qui renvoie le polynôme ainsi créé.

Question 3) Écrire la procédure *afficherPolynome*, prenant en paramètre un polynôme d'ordre 2 et l'affichant à l'écran.

Question 4) Écrire la fonction *calculerValeur*, prenant en paramètre un polynôme d'ordre 2 (noté *P*) et un réel (noté *x*), et retournant la valeur *P(x)* pour ce *x* donné.

Question 5) Écrire la procédure *deriverPolynome*, prenant en paramètre un polynôme d'ordre 2 et qui calcule sa dérivée (elle modifie le polynôme passé en paramètres).

Rappel : La puissance et la racine carré sont définis en C par les fonctions *pow* et *sqrt* inclus dans le fichier *math.h*.

2. Gestion des dates (0.5 + 0.5 + 2 + 2 + 2 = 7)

Question 1) Définir l'énumération *Mois* composée des mois de l'année, de Janvier à Décembre.

Question 2) Définir la structure *Date* qui est composée d'un numéro de jour, d'une variable de type *Mois*, et du numéro de l'année.

Question 3) Écrire une fonction *saisirMois* demandant à l'utilisateur d'entrer une chaîne de caractères et qui renvoie une variable de type *Mois*. Pour cela, vous pouvez parcourir le tableau **tab** suivant, et vérifier si la chaîne entrée par l'utilisateur est la même que celle de la *i*^{ème} case du tableau, afin de récupérer l'indice *i*

Rédigé par : Elizabeth R., Vincent G. et Rachid C.

Ref : *ING1-PROGC-EXAMEN*

A l'intention de : Etudiants des ING1

Créé le : 10/01/2012

correspondant au mois dans le tableau. La fonction `int strcmp(char*, char*)` définie dans `string.h`, renvoie 0 si les 2 chaînes passées en paramètres sont égales.

```
char* tab[12] = {"Janvier", "Fevrier", "Mars", "Avril", "Mai", "Juin", "Juillet", "Aout", "Septembre",  
               "Octobre", "Novembre", "Decembre"};
```

Question 4) Écrire une fonction `saisirDate` demandant une date au clavier et renvoyant une structure `Date`. Pour simplifier, on ne prendra pas en compte la limite du nombre de jours pour un mois (l'utilisateur peut entrer n'importe quelle valeur entière).

Question 5) Écrire une procédure permettant d'afficher à l'écran une variable de type `Date` sous la forme :
jj/mm/aaaa

(Ne pas oublier le 0 si le jour est inférieur à 10)

Question 6) Écrire une fonction `comparerDates` prenant 2 dates `d1` et `d2` en paramètres, et retournant -1 si `d1` est avant `d2`, 1 si `d1` est après `d2` et 0 si les dates sont égales.

3. Gestion des dates (dynamique) (0.5 + 1.5 + 1.5 + 3 = 6.5)

Pour cet exercice, nous reprenons la définition des `Date` définies à l'exercice précédent.

Question 1) Définir la structure `TableauDate` composée d'un **tableau dynamique** de `Date`, ainsi que de la taille de ce tableau.

Question 2) Écrire une fonction `saisirTableauDate`, permettant de saisir complètement, et retourner une variable de type `TableauDate`

Question 3) Écrire une procédure `enregistrerTableauDate`, permettant de stocker la variable de type `TableauDate` dans un fichier nommé « `sauvegarde.dat` »

Question 4) Écrire une procédure prenant en paramètre une variable de type `TableauDate` et le triant par date croissante (vous utiliserez l'un des tris vus en TP, que vous adapterez à la question, seule l'adaptation du tri rapportera des points).