

HAE303 - LANGAGE C

Durée : 1h.20

- Exercice : Programme à Corriger -

I.1 - Préparation

1. Créez un répertoire dont le titre est votre nom et prénom, au format : NOM_Prenom. Par exemple : DUPONT_Pierre.
2. Placez vous dans ce répertoire avec le terminal. Vous écrirez la totalité de vos codes à l'intérieur de ce répertoire.
3. Vous devrez transmettre ce répertoire à l'enseignant, en suivant une procédure qui vous sera précisée à la fin de l'épreuve.

Toute copie qui sera rendue sans suivre cette procédure vaudra 0/20.

I.2 - Compilation

On donne le code ci-dessous. Téléchargez-le (*ne faites PAS un copier/coller à partir de ce document*) et copiez le code dans un fichier que vous appellerez "Q1.cpp". Ensuite, compilez le, constatez que le compilateur mentionne des erreurs, et corrigez ces erreurs.

Dans cette phase, on ne demande de corriger QUE les erreurs indiquées par le compilateurs : on ne cherchera pas à faire en sorte que le programme s'exécute correctement, on a juste besoin qu'il se compile.

```
1  #include <vector>
2  using namespace std;
3
4  typedef struct {
5      double coeff;
6      double exp;
7  } terme;
8
9
10 int main()
11 {
12
13     int n=1;
14     vector<terme> Polynome1;
15
16     Polynome1.resize(10);
17     Polynome1[0].coeff = 1.0;
18     Polynome1[0].exp = 1.0;
19
20     // creation d'un polynome
```

```

21     den = 1.0;
22     while(n<100) {
23         den = den*(2*n+1.0)*(2*n)*(-1.0)
24         Polynome1[n].coeff = 1/den;
25         Polynome1[n].exp    = 2.0*n+1.0;
26         n = n + 1;
27     }
28
29 // affichage du polynome
30     for(n = 0; n < size() ; n++) {
31         cout << Polynome1[n].coeff << "x^" << Polynome1[n].exp << " + ";
32     }
33
34     return 0;
35 }

```

I.3 - Exécution

Copiez/collez le code obtenu à la fin de la question précédente dans un nouveau fichier **Q2.cpp**, et travaillez sur ce nouveau fichier pour cette question.

Si vous lancez le programme, vous constatez qu'il plante, il n'arrive pas jusqu'au bout. Corrigez l'erreur, qui se situe **après la ligne 19**. Il y a une seule erreur. Toute correction avant la ligne 19 sera hors sujet.

I.4 - Nouveau Polynome

Copiez/collez le code obtenu à la fin de la question précédente dans un nouveau fichier **Q3.cpp**, et travaillez sur ce nouveau fichier pour cette question.

Vous devez maintenant comprendre ce que fait le code afin de pouvoir le modifier.

À partir de la ligne 33, écrivez un code qui crée un nouveau Polynome, que vous appelez Polynome2, qui sera le codage du polynome simple :

$$3x^2 + 5x - 5$$

Puis affichez le.

I.5 - Valeur du Polynome

Copiez/collez le code obtenu à la fin de la question précédente dans un nouveau fichier **Q4.cpp**, et travaillez sur ce nouveau fichier pour cette question.

À la fin du main, juste avant le return 0, écrivez un morceau de code qui calcule la valeur du polynome Polynome1 pour $x = 1$, et qui affiche le résultat.

Vous aurez besoin de la fonction pow de la bibliothèque cmath, qui s'utilise comme suit :
l'expression pow(a,b); renvoie le résultat de a^b

I.6 - Fonctions

Copiez/collez le code obtenu à la fin de la question précédente dans un nouveau fichier **Q5.cpp**, et travaillez sur ce nouveau fichier pour cette question.

En vous inspirant de portions du main, créez les fonctions dont les prototypes sont données ci-dessous :

- vector<terme> creerPolynome1(int N); qui renvoie un polynome identique au Polynome1 du main, mais pour un nombre de termes décidé par la valeur du paramètre N.
- void affichePolynome(vector<terme> P); qui affiche le polynome P passé en paramètre.

— `double valeurPolynome(vector<terme>P, double x)` qui renvoie la valeur du polynome pour la valeur de `x` passée en paramètre.

Pour terminer, utilisez ces fonctions dans le `main` pour remplacer la partie du `main` qui est équivalente, à chaque fois que c'est possible.

1.7 - Aller plus loin (difficile)

*Copiez/collez le code obtenu à la fin de la question précédente dans un nouveau fichier **Q6.cpp**, et travaillez sur ce nouveau fichier pour cette question.*

Ecrivez une fonction :

`vector<terme> produitPolynome(vector<terme> P1, vector<terme> P2);` qui renvoie le polynome qui est le produit de `P1` et `P2`. Par exemple, si on donne :

$P1(x) = 3x + 2$ et $P2(x) = x^2 + x + 1$, la fonction doit renvoyer le polynome qui est le résultat du calcul :

$$P(x) = (3x + 2) \times (x^2 + x + 1)$$

c'est à dire :

$$P(x) = 3x^3 + 3x^2 + 3x + 2x^2 + 2x + 2$$

On **ne demande pas** de donner $P(x)$ sous sa forme simplifiée, dans laquelle chaque x^n serait représenté une seule fois avec le bon coefficient.

Pour terminer, faites appel à cette fonction pour calculer le produit de `Polynome1` et `Polynome2` du `main` et affiche le polynome résultant.

*Pour rendre votre copie : créez une archive en cliquant avec le bouton droit sur le dossier **NOM_Prenom** que vous avez créé au début de l'épreuve, puis en choisissant "Créer une archive" dans le menu, et suivre pour obtenir un fichier .zip. Vous enverrez ce fichier par internet selon la procédure indiquée à la fin de l'épreuve.*