

Seul document autorisé : une feuille A4 recto-verso écrite de votre main.

La notation tiendra compte de la présentation et de la clarté de la rédaction. La barème est indicatif et pourra être modifié si nécessaire.

★ Exercice 1: Calcul Intégral (8pts)

Nous souhaitons écrire un programme permettant de calculer l'intégrale d'un polynôme à coefficient réel sur un intervalle $[a, b]$. Pour cela, chaque polynôme sera défini par son degré et par un tableau correspondant à ses coefficients. Voici deux exemples pour les tableaux de coefficients :

$T = [1, 0, 5, 3]$ représente le polynôme $P = 1 + 5X^2 + 3X^3$

$U = [0, 1, 0, 1, 0, 1]$ représente le polynôme $P = X + X^3 + X^5$

▷ **Question 1:** (0,5pt) Écrivez une structure de données `poly` contenant toutes les informations relatives à un polynôme.

▷ **Question 2:** (1,5pts) Écrivez une fonction `poly_new()`, prenant en arguments les informations initiales et retournant un pointeur vers une nouvelle structure `poly`.

▷ **Question 3:** (0,5pt) Écrivez une fonction `poly_free()`, prenant en arguments une référence à un polynôme pour le détruire en libérant la mémoire qu'il occupait.

Rappel : Soit un polynôme P de degré n tel que $P(x) = \sum_{i=0}^n a_i * X^i$ alors la primitive Q du polynôme P s'annulant en 0 s'écrit $Q(x) = \sum_{i=0}^n \frac{a_i}{i+1} * X^{i+1}$

▷ **Question 4:** (1,5pts) Écrivez une fonction `primit()`, prenant en argument un polynôme (une structure `poly`) et retournant un pointeur vers sa primitive (une nouvelle structure `poly`).

▷ **Question 5:** (1,5pts) Écrivez la fonction `valeur_poly()` prenant en argument un polynôme P (une structure `poly`) et un nombre t et retournant la valeur $P(t)$.

▷ **Question 6:** (2,5pts) En reprenant les fonctions précédentes, écrivez le programme, compilé sous le nom `integrale` permettant de calculer l'intégrale d'un polynôme P entre deux valeurs a et b . La commande `./integrale 3 0.2 2.0 3.1 6.0 5.0 10.0` permettra de calculer l'intégrale $\int_5^{10} 0.2 + 2x + 3.1x^2 + 6x^3 dx$.

★ Exercice 2: La commande strings (3pts)

La commande `strings` permet d'étudier les parties lisibles d'un fichier binaire dont le nom est passé en argument. Elle n'affiche à l'écran que les caractères du fichier dont les codes Ascii sont dans l'intervalle $[32, 126]$ (ce sont les caractères "affichables", les autres étant des codes de contrôle).

Exemple : si `fichier` contient les caractères de codes Ascii (12, 33, 65, 69, 5, 4, 77, 89, 14, 35, 101, 102) alors `strings fichier` affichera les caractères : (33, 65, 69, 77, 89, 35, 101, 102). Soit : `!AEMY#ef`

▷ **Question 1:** (3pts) Écrivez en C la commande `strings`.

★ Exercice 3: Compter les lettres (5pts)

Nous souhaitons écrire une fonction qui compte le nombre d'occurrence de caractères dans une chaîne donnée.

▷ **Question 1:** (2pts) Écrivez la fonction `int compte_un(char *chaine, char car)` comptant le nombre d'occurrences du caractère passé en deuxième argument dans la chaîne passée en premier argument. Ainsi, `compte_un("bonjour", b)` doit retourner 1 tandis que `compte_un("bonjour", o)` doit retourner 2.

▷ **Question 2:** (3pts) Écrivez la fonction `int *compte_tous(char *chaine)` qui retourne un tableau de 26 entiers (que vous devrez allouer), chacun indiquant le nombre d'occurrences de la lettre correspondante.

★ **Exercice 4: Lecture de code (6pts)** Commentez les étapes de ce programme en décrivant le contenu des différentes variables (faire un schéma des tableaux et des variables aux points demandés). Indiquez également ce qu'il imprime.

```
#include <stdio.h>

int main(){

    char mot[] = "vous aimez";
    char mot2[7];
    char *ptr,**ptr2;
    int i = 0;

    ptr = mot+3;
    *ptr = *mot;
    *mot = *mot-2;
    mot[2] = *(mot+5);

    /* Faire un schéma à ce point */

    ptr = ptr - 2;
    *ptr = 'r';
    ptr2 = &ptr;
    for(i=0;i<3;i++) {
        mot[i+4]=mot[i+5];
    }

    /* Faire un schéma à ce point */

    ptr = ptr+5;
    --*ptr;
    --(*ptr2+1);
    *(mot+9) = 'r';

    /* Faire un schéma à ce point */

    for(i=0;i<7;i++) {
        mot2[i] = mot[i];
    }

    /* Faire un schéma à ce point */

    printf("Choisissez un %s que vous aimez et vous n'aurez pas à %s un seul jour
de votre vie. (Confucius)", mot2, mot);
    return 0;
}
```