Bibliothèque de Polynômes

On souhaite écrire une petite bibliothèque pour manipuler des polynômes à coefficients entiers.

1. Créer un fichier d'en-tête `poly.h` contenant :
   - une garde d'inclusion ;
   - la déclaration d'une fonction `int poly_eval(int* coeffs, int deg, int x)` qui calcule la valeur du polynôme de degré `deg` et de coefficients `coeffs[0], ..., coeffs[deg]` en `x` (utiliser la méthode de Horner) ;
   - la déclaration d'une fonction `void poly_derive(int* coeffs, int deg, int* result)` qui calcule les coefficients du polynôme dérivé et les stocke dans `result`.
2. Écrire un fichier `poly.c` qui inclut `"poly.h"` et définit les deux fonctions.
3. Écrire un fichier `main.c` qui :
   - lit depuis la ligne de commande le degré n d'un polynôme ;
   - lit ensuite $n+1$ entiers correspondant aux coefficients ;
   - affiche la valeur du polynôme en $x = 2$;
   - affiche les coefficients de son polynôme dérivé.
4. Donner les commandes de compilation séparée permettant d'obtenir un exécutable `poly-main`.

Matthieu Boyer
Planche B

Bibliothèque de Matrices

On souhaite concevoir une bibliothèque `matrix` pour manipuler des matrices d'entiers $n \times n$.

1. Écrire un fichier d'en-tête `matrix.h` qui :
   - déclare une constante `MAX_SIZE` valant 10 ;
   - déclare un type `typedef int Matrix[MAX_SIZE][MAX_SIZE]` ;
   - déclare les fonctions suivantes :
     - `void matrix_init(Matrix m, int n)` ;
     - `void matrix_add(Matrix a, Matrix b, Matrix result, int n)` ;
     - `void matrix_mult(Matrix a, Matrix b, Matrix result, int n)` ;
     - `void matrix_print(Matrix m, int n)`.
2. Proposer, en commentaire, ce que fait chaque fonction.
3. Écrire un fichier `matrix.c` qui inclut `"matrix.h"` et définit les fonctions (on pourra laisser les corps vides dans un premier temps pour `matrix_init`).
4. Écrire un fichier `main.c` qui utilise la bibliothèque :
   - crée deux matrices A et B de taille 3 ;
   - les initialise depuis l'entrée standard ;
   - calcule et affiche $A + B$;
   - calcule et affiche $A \times B$.

Bibliothèque de Chaînes de Caractères

On souhaite concevoir une bibliothèque `string` pour manipuler des chaînes de caractères.

1. Écrire un fichier d'en-tête `string.h` qui déclare les fonctions suivantes :
   - `int string_length(char* s);`
   - `void string_copy(char* src, char* dest);`
   - `void string_concat(char* s1, char* s2, char* result);`
   - `int string_compare(char* s1, char* s2);`
   - `void string_reverse(char* s)`.
2. Proposer, en commentaire, une description de chaque fonction (par exemple `string_compare` renvoie un entier négatif, nul ou positif selon que `s1` est lexicographiquement avant, égale ou après `s2`).
3. Écrire un fichier `string.c` qui inclut `"string.h"` et définit les fonctions (on pourra laisser les corps vides dans un premier temps).
4. Écrire un fichier `main.c` qui utilise la bibliothèque :
   - lit deux chaînes de caractères depuis l'entrée standard ;
   - affiche leur longueur respective ;
   - affiche leur concaténation ;
   - affiche laquelle est avant l'autre dans l'ordre lexicographique ;
   - affiche chaque chaîne inversée.