

Travaux Dirigés de Synthèse
PROGRAMMATION ORIENTEE OBJET EN LANGAGE C++

PROBLEME :

On désire écrire une classe *Suite* définie de la manière suivante :

$$\begin{cases} U_0 = 1 \\ U_n = U_{n-1} * q \end{cases}$$

Avec q, la raison de la suite.

I- Dans cette classe prévoir :

- 1- Un membre entier privé nommé (**Nb**) constituant le nombre d'éléments de la suite,
- 2- Un tableau de réels **dynamique** privé nommé (**S**) dans lequel on va ranger les éléments de la suite,
- 3- Un constructeur qui calcule (N) termes de la suite de raison (q) et les range dans le tableau membre (S).
- 4- Un constructeur par copie, qui sera appelé lors de l'exécution de l'expression exemple suivante :

```
Suite a(3);  
Suite b = a;
```

- 5- Un destructeur qui libère l'espace mémoire réservé pour le membre (S).
- 6- Une fonction membre « **Affiche** », permettant d'afficher les éléments de la suite.
- 7- Une fonction membre « **Meme** » permettant de retourner 1 si la suite en cours est la même que celle reçue en argument et 0 dans le cas contraire,
- 8- Une fonction membre « **Opposée** », permettant de retourner une suite constituée des éléments opposés de ceux de la suite en cours,
- 9- Une fonction membre « **Homo_Suite** », permettant de multiplier tous l'élément de la suite en cours par un nombre réel reçu en argument et de retourner la suite résultante,
- 10- Une fonction membre « **Produit_Suite** », permettant de multiplier les éléments de la suite en cours par leurs homologues (1 avec 1, 2 avec 2, ...etc.) d'une autre suite reçue en argument et de retourner la suite résultante,

- 11- Une fonction membre « **Prod_Scal_Suite** », (*une sorte de produit scalaire de deux suites*), permettant de multiplier les éléments de la suite en cours par leurs homologues (1 avec 1, 2 avec 2, ...etc.) d'une autre suite reçue en argument, de faire la somme de ces produits et de retourner le nombre réel résultat,
- 12- Une fonction membre « **Somme_Suite** », permettant calculer la somme des deux suites, en sommant les éléments de la suite en cours par leurs homologues (1 avec 1, 2 avec 2, ...etc.) d'une autre suite reçue en argument et de retourner suite résultante,
- 13- Une fonction membre de sur-définition de l'opérateur somme (+) qui permettra d'écrire dans un programme l'expression exemple suivante :

```
suite S1, S2, S3;  
S3 = S2 + S1;
```

- 14- Une fonction membre permettant de créer l'opérateur « produit scalaire de deux suite, comme défini ci-dessus (Question 11) », sur-définition de l'opérateur (*), c'est à dire de donner une signification à l'opération exemple suivante :

```
Suite S1, S2;  
float PS;  
PS = v1 * v2;
```

- 15- Une fonction membre « multiplication des éléments de la suite par un réel, comme défini ci-dessus (Question 9) » (une autre sur-définition de l'opérateur *) permettant de donner une signification au produit d'un réel et d'une suite, c'est à dire de donner une signification à l'opération exemple suivante :

```
Suite S1, S2;  
float h;  
S2 = S1 * h ;
```

II- Ecrire une fonction indépendante amie à la classe « Suite » nommée « Coincide » qui fait le même travail que la fonction membre « meme » définit précédemment dans la Question 7, c'est-à-dire déterminer la coïncidence deux objets de la classe Suite.

III- Prévoir fonction main dans laquelle, vous allez mettre en œuvre toutes les fonctions décrites ci-dessus ainsi que les différentes opérations correspondantes demandées.