

## TP 3

### Caractéristiques du langage Fortran :

#### 1) le type « complexe » :

Puisque Le langage Fortran est conçu pour être utilisé en mathématiques et dans les applications de calcul scientifique, il fournit un type standard pour les nombres complexes. Une valeur du type complexe est représentée par un couple de nombres réels. Le premier nombre correspond à la partie réelle du nombre complexe et le deuxième nombre correspond à la partie imaginaire du même nombre complexe. Une valeur du type complexe est alors écrite comme suit : **(5,2)** ce qui correspond au nombre: **5 + 2i**

### Travaux Pratiques :

Ecrivez le programme suivant à l'aide du compilateur Salford FTN95, puis exécutez le.

```

program nouveau_type
  implicit none
  complex A,B
  A = (4, 2.5)
  B = (10.5,3)
  print *, '(4+ 2.5i) + (10.5 + 3i) =', A +B
  print *, '(4+ 2.5i) * (10.5 + 3i) =', A * B
end program
  
```

### Exercice1

Traduisez les algorithmes suivants en **programmes Fortran**, sauvegardez ces derniers dans des fichiers sources (.f95), puis lancez la compilation, l'édition de liens, et l'exécution de chaque programme.

Algorithme sommes Variable i, j, s1 : entier s2 : réel Début  s1 ← 0 s2 ← 0 Pour i de 1 à 100 faire s1 ← s1 + i Finpour Pour j de 1 à 50 faire s2 ← s2 + 1.0/j Finpour  Ecrire (s1) Ecrire (s2) Fin	Algorithme calcul Variable v, c : entier Début  Lire(v) Ecrire('donnez un nombre entier positif') Lire (v) Tantque (v < 0) faire Ecrire ('donnez un nombre entier positif') Lire (v) FinTantQue  c ← v * v Ecrire (c) Fin
--	--

### Exercice2

1. Ecrire un programme pour lire le rayon d'un cercle au clavier et renvoyer l'aire et le volume du cercle et de la sphère correspondants (rappel :  $\pi r^2$  et  $4/3 \pi r^3$ ).
2. Compiler, tester.
3. Rajouter une boucle infinie do dans laquelle, après l'affichage, on demande à l'utilisateur s'il désire continuer ou pas. Si la réponse est oui, on demande à nouveau

un rayon, on recalcule l'aire et le volume, on affiche, et on repose la question. Si la réponse est non, le programme s'arrête.

### **Exercice3**

Écrire un algorithme qui demande un nombre entier  $N$ , et qui calcule la somme :

$$s = \sum_{i=1}^n i = 1 + 2 + 3 + 4 + \dots + n$$

Résoudre le problème avec la boucle TANTQUE et POUR

### **Traduire ces algorithmes en programmes Fortran**

### **Exercice4**

Écrire un algorithme somme qui calcule :

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!},$$

Tels que  $x$  est un réel et  $n$  et un entier

### **Traduire ces algorithmes en programmes Fortran**

### **Exercice5**

Écrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée 8 !, vaut

$$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$$

Résoudre le problème avec la boucle TANTQUE et POUR

### **Traduire ces algorithmes en programmes Fortran**

### **Exercices facultatifs**

### **Exercice6**

1. En utilisant les boucles, Écrire un algorithme qui calcule  $x^y$  tel que  $x$  est un réel et  $y$  est un entier.

Exemple :  $7^5 = 7 \times 7 \times 7 \times 7 \times 7 = 16807$

2. Écrire le même algorithme sans utiliser les boucles

### **Traduire ces algorithmes en programmes Fortran**

### **Exercice7**

Écrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$