

TP1

Exercice1:

Ecrire un algorithme qui permet de calculer le carré d'un nombre entier.

Exercice2:

Ecrire un algorithme qui permet de calculer $F(x, y)$ tel que x et y sont des nombres réels et :

$$F(x,y) = \frac{x^2 + y}{5}$$

Travaux pratiques:

1) Le langage Fortran:

Fortran est un langage de programmation utilisé principalement en mathématiques et dans les applications de calcul scientifique. Il a été créé par le scientifique américain **John Backus** entre 1954 et 1957. Le nom du langage vient de la contraction de **FOR**mula **TRAN**slator qui signifie traducteur de formules. Aujourd'hui encore le langage Fortran reste très utilisé en raison de la présence de très nombreuses bibliothèques de fonctions utilisables dans les applications de calcul scientifique.

Plusieurs versions du langage ont vu le jour, à savoir: Fortran 77, **Fortran 90**, Fortran 95, et Fortran 2003. Chaque version contient un bon nombre de nouvelles fonctionnalités mais reste compatible avec les versions antérieures.

Puisque l'ordinateur ne manipule que des informations binaires, les programmes doivent être écrits en binaire (**langage machine**) pour être exécutés. Cependant, il n'est pas facile pour nous d'écrire un programme en binaire (suite de 0 et de 1), pour cette raison nous utilisons des langages de programmation évolués (exemple: Fortran, Pascal, JAVA, C++, ...).

La traduction d'un programme écrit dans un langage évolué vers le langage machine est effectuée par un logiciel appelé **compilateur**. Pour le langage Fortran il existe plusieurs compilateurs sur le marché comme: **Salford FTN95**, Intel Fortran Compiler, Absoft Fortran Pro, IBM Fortran 95 compiler, Lahey/Fujitsu Fortran 95 compiler, ...

Trois types de fichiers interviennent dans l'élaboration d'un programme Fortran :

- Le fichier **source** (extension .F90) écrit par le programmeur.
- Le fichier **objet** (extension .OBJ) généré par le compilateur.
- Le fichier **exécutable** (extension .EXE) généré par l'éditeur de liens.

Le programmeur écrit le fichier source contenant le programme Fortran. Le compilateur traduit ce dernier en langage machine et sauvegarde le résultat dans le fichier objet. Si le fichier objet contient des appels vers d'autres fichiers objets, l'éditeur de liens se chargera alors de les relier dans un fichier exécutable.

2) Utilisation du compilateur Salford FTN95:

- a) Créer un dossier sur le bureau.
- b) Lancer l'environnement de programmation « Salford Plato IDE ».
- c) Créer un fichier source (File->New), puis le sauvegarder dans le dossier créé précédemment).
- d) Ecrire le programme suivant qui permet de faire l'addition de deux nombres:

```
program addition
implicit none
real :: answer, x, y
print*, 'Enter two numbers'
read*, x
read*, y
answer=x+y
print*, 'the total is:', answer
end program addition
```

- e) Lancer la compilation (Project->Compile).
- f) Lancer l'édition de liens (Project->Build).
- g) Lancer l'exécution (Project->Run).
- h) Voir les fichiers: «.f95», «.obj», «.exe» dans le dossier précédent.

3) Traduction d'algorithmes en programme Fortran:

Le tableau suivant contient les instructions Fortran équivalentes aux mots-clés utilisés dans les algorithmes :

Algorithme	PROGRAM
Entier	INTEGER
Réel	REAL
←	=
Lire	READ *,
Ecrire	PRINT *,
Fin	ENDPROGRAM

- a) A l'aide de ce tableau traduisez les algorithmes des exercices 1 et 2 en programmes Fortran et sauvegardez ces derniers dans des fichiers sources .f90 ou f95, puis lancez la compilation, l'édition de liens, et l'exécution de chaque programme.

(Exercices facultatifs)

Exercice 3:

On dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).

Traduire cet algorithme en fortran.

Exercice 4

Ecrire un algorithme qui convertit en Octets, Kilo octets, Mega octets et Giga octets un nombre donnée en bits.

Traduire cet algorithme en fortran.

Exercice 5

Ecrire un algorithme qui lit au clavier l'heure et les minutes et qui affiche l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21, puis 30, l'algorithme doit répondre : "Dans une minute, il sera 21 heure(s), 31 minute(s)".

Traduire cet algorithme en fortran.