

TD n°5'

Créer un sous-dossier *TD_05'* dans le dossier *TD_PROG_IMP*. La compilation des programmes sera faite en utilisant un fichier *Makefile* (fourni avec le sujet) et trois fichiers « *m_type.f90* », « *main.f90* » et « *sousprog.f90* » contenant respectivement un module avec des structures, le programme principal et les sousroutines/fonctions qui seront développées pendant le TD.

EXERCICE 1 : Programme « coucou » avec un fichier *Makefile*

1°) Dans le fichier « *sousprog.f90* », créer la subroutine appelée *affiche_coucou* affichant à l'écran la chaîne de caractères « Coucou ».

2°) Dans le fichier « *prog.f90* », écrire le programme principal appelant la subroutine *affiche_coucou*.

3°) Modifier, si nécessaire, le fichier *Makefile* afin de pouvoir compiler les fichiers « *prog.f90* » et « *sousprog.f90* ». Dans le dossier contenant les deux fichiers *.f90 et le fichier *Makefile*, compiler le code (via la commande : *make*) et exécuter le programme.

EXERCICE 2 : Recherche et traitement à partir d'une base de données

On dispose d'un fichier formaté appelé « *amis.dat* » (fourni avec le sujet) contenant une liste d'informations (numéro, prénom, date de naissance). L'objectif du programme à développer est de rechercher dans cette base de données, s'il y a une ou plusieurs personnes dont c'est l'anniversaire aujourd'hui.

1°) Dans le fichier « *m_type.f90* », créer un module appelé *m_type*. Dans ce module, définir une structure de données (au sens du Fortran 90) appelée *personne* et ayant pour champs : un entier noté *num*, une chaîne de 30 caractères appelée *nom* et un tableau de 3 entiers appelé *jma*. Préciser en commentaires l'utilité de ces champs.

2°) Dans le fichier « *sousprog.f90* », créer la fonction *test_annif* de type booléen (logique) qui est vraie lorsque quatre entiers notés j_1, j_2, m_1 et m_2 sont tels que $j_1=j_2$ et $m_1=m_2$, fausse sinon.

3°) Dans le fichier « *prog.f90* », créer un programme principal qui demande à l'utilisateur les valeurs de j_1, j_2, m_1 et m_2 et affiche à l'écran un message lorsque *test_annif* est vraie.

On cherche maintenant à déterminer le nombre d'amis présent dans la base de données. Pour cela, on utilise la subroutine *calcul_nb_lignes* suivante, qui exploite la fonctionnalité 'iostat' du Fortran 90, telle que :

```

subroutine calcul_nb_lignes(nom_fichier,nlignes)

  implicit none
  ! DECLARATIONS
  ! Variables en argument
  character (len=15), intent(in) :: nom_fichier
  integer, intent(out)          :: nlignes
  ! Variables locales
  integer :: ierr ! =0 si la ligne n'est pas vide, /=0 sinon

  ! INSTRUCTIONS
  open(10,file=nom_fichier,iostat=ierr)
  nlignes = 0
  do while (ierr==0)
    read(10,*,iostat=ierr)
    if (ierr==0) then
      nlignes = nlignes + 1
    end if
  end do
  close(10)

end subroutine calcul_nb_lignes

```

4°) Dans le fichier « sousprog.f90 », implémenter la subroutine *calcul_nb_lignes*.

5°) Ecrire, dans le programme principal, les instructions permettant de calculer le nombre d'amis du fichier « amis.dat » et d'afficher sa valeur à l'écran.

On cherche à présent à déterminer la date du jour. Pour cela, on utilise la subroutine intrinsèque à Fortran 90 appelée *idate(date_du_jour)* et qui possède un seul argument de sortie : un vecteur de 3 entiers (appelé ici *date_du_jour*) contenant respectivement le jour, le mois et l'année.

6°) Appeler dans le programme principal la subroutine *idate* et faire afficher la date du jour à l'écran.

Dans un dernier temps, on souhaite balayer la base de données et afficher à l'écran le message suivant, lorsque qu'un anniversaire se présente : « Aujourd'hui, c'est l'anniversaire de ... qui a ... ans ».

7°) Dans le fichier « sousprog.f90 », créer la subroutine appelée *qui_annif* qui remplit cette fonctionnalité, en utilisant la structure *personne*. Indication : cette subroutine consiste à : ouvrir le fichier « amis.dat » ; balayer le fichier pour récupérer le numéro, nom et date de

naissance de chaque amis (en utilisant la structure *personne*) ; tester la correspondance entre la date de naissance et la date du jour ; afficher à l'écran l'occurrence d'un anniversaire.

8°) Modifier le programme principal, de sorte à faire appel à la subroutine *qui_annif*.

Pour aller plus loin :

On souhaite pouvoir faire quelques statistiques avec cette base de données (calcul de l'âge moyen, l'âge minimum et maximum des amis). Pour cela, on utilise la structure *personne* pour créer un vecteur dont le nombre d'éléments est le nombre total d'amis et le type d'éléments est du type *personne*.

9°) Dans le programme principal : (1) créer la structure *personne*, identique à celle de la question n°1 ; (2) déclarer un vecteur noté *p* dont les éléments sont du type *personne*, en utilisant l'allocation dynamique ; (3) allouer, lorsque le nombre d'amis est connu, le vecteur *p* ; (4) dé-allouer *p* en fin de programme.

10°) Modifier la subroutine *qui_annif* afin qu'elle permette en plus de récupérer l'ensemble des informations de la base de données dans *p*. Indication : le vecteur *p* doit à présent être mis en argument de la subroutine.

11°) Dans le fichier « sousprog.f90 », créer la subroutine appelée *age_stat* qui (1) calcule l'âge de chaque ami ; affiche à l'écran (2) l'âge minimum, (3) l'âge maximum et (4) l'âge moyen.

12°) Dans le programme principal, appeler la subroutine *age_stat*.