

SNT4U16 - Initiation à la programmation

Licence SNTE 2^{ème} année 2013-2014

Session 1 - CORRIGÉ

mercredi 07 mai 2014

DURÉE : 2 HEURES

Le barème est donné à titre indicatif

Aucun document n'est autorisé

PRÉAMBULE : À LIRE ABSOLUMENT!!!

▷ Comptes d'examen

Pour l'examen, vous vous identifierez sur les machines avec vos login et mot de passe habituels. Vous serez alors connecté sur votre "compte d'examen". Celui-ci contient un répertoire `exam/` dans lequel vous devrez travailler. Ce répertoire `exam/` contient lui-même des sous-répertoires correspondant aux différents exercices, au sein desquels vous trouverez les fichiers source fortran pré-crés.

▷ Fichiers à compléter

- Une grande partie des réponses aux questions seront données sous forme de fichiers Fortran 90 qui seront relevés sur vos comptes d'examen de manière *automatique* à la fin de l'épreuve.

- Pour chaque exercice et pour chaque question, il est indiqué si la réponse doit être donnée sous forme de fichier et dans ce cas, le nom du fichier est précisé. **Tous les fichiers à rendre ont déjà été créés sur votre compte. Vous devrez simplement les éditer et les compléter (à l'aide de kate par exemple) sans changer leur nom ni leur emplacement.** Les fichiers-réponse sont répartis dans des répertoires dont les noms correspondent aux numéros d'exercice.

- Vous compilerez vos fichiers et les exécuterez pour vérifier la justesse de vos programmes. La bonne compilation et la bonne exécution des programmes constitueront une part importante de l'évaluation.

- Pour certaines questions, lorsque ce sera demandé, vous aurez à indiquer sur papier le résultat de l'exécution.

- Avant la fin de l'examen, vérifiez bien que tous vos fichiers sont sauvegardés sur votre compte d'examen, au bon emplacement et sous le bon nom. **Seuls les fichiers correctement nommés et placés à l'endroit approprié seront relevés et corrigés.**

▷ Réponses sur copie

Lorsque rien n'est précisé, vous répondrez aux questions sur copie.

▷ Formulaires

Un "pense-bête" *Fortran 90 et commandes linux* est mis à votre disposition.

▷ Ordre des exercices

Les exercices sont indépendants et peuvent être traités dans l'ordre souhaité.

▷ Lecture attentive des questions

Il n'est parfois pas inutile pour certains de rappeler que les questions doivent être lues en entier et avec concentration. Elles sont posées de manière précise et, autant que possible, non ambiguë. "Gagner du temps" en lisant les questions en diagonale vous expose à répondre à autre chose que ce qui est demandé, donc à ne pas être noté sur votre réponse et donc au final à **perdre** un temps précieux . . .

▷ Barème indicatif

La notation totale compte sur 20 points, avec 3,5 points "bonus" supplémentaires. Le barème indiqué n'est pas définitif et est susceptible d'être modifié.

1) (4 pts) Lettres

Écrire un programme Fortran 90 qui demande à l'utilisateur d'entrer une phrase au clavier, et qui compte et affiche à l'écran le nombre total de lettres non accentuées (minuscules ou majuscules) utilisées dans la phrase.

Pour répondre à la question, le fichier `~/exam/exo1/exo1.f90` devra être complété.

```
1 PROGRAM lettres
2
3 IMPLICIT NONE
4
5 CHARACTER (len=2000) :: phrase
6 INTEGER :: compteur, ic, i
7
8 WRITE (*,*) "Entrez une phrase entre guillemets SVP:"
9 READ (*,*) phrase
10 ! Ou alors on utilise le format '(a)' pour la lecture
11
12 compteur=0
13 DO i=1,LEN(TRIM(phrase))
14   ic = ICHAR(phrase(i:i))
15   IF ((ic >= ICHAR("A") .AND. ic <= ICHAR("Z")) .OR. (ic >= ICHAR("a") .AND. ic <= ICHAR("z"))) THEN
16     compteur = compteur + 1
17   END IF
18 END DO
19
20 WRITE (*,*) "Votre phrase comporte", compteur, " lettres non accentuées."
21
22 END PROGRAM lettres
```

2) (7,5 pts) Nombres premiers

a) (1,5 pts) programme mystère

On donne le programme Fortran 90 suivant :

```
1 PROGRAM mystere
2
3 IMPLICIT NONE
4
5 INTEGER :: x,y
6
7 WRITE (*,*) "Entrez un entier x"
8 READ (*,*) x
9 WRITE (*,*) "Entrez un entier y"
10 READ (*,*) y
11
12 IF (x/y == x/(y*1.0)) THEN
13   WRITE (*,*) ...
14 ELSE
15   WRITE (*,*) ...
16 END IF
17
18 END PROGRAM mystere
```

Compléter les lignes 13 et 15 en remplaçant de manière adéquate les pointillés. A quoi sert ce programme? (vous justifierez votre réponse)

Ce programme compare les valeurs de x/y et $x/(y*1.0)$. Comme x et y sont de type `INTEGER`, la première des deux divisions est une division entière. Le résultat de x/y est la valeur entière de $\frac{x}{y}$. Dans le cas de la deuxième division, ($y*1.0$) est maintenant de type `REAL`, de sorte que la division $x/(y*1.0)$ est une division réelle. Si les deux résultats coïncident, cela veut dire que la division de x par y "tombe juste" et donc que x est divisible par y .

```
13 write (*,*) x, "est divisible par", y
```

```
15 write (*,*) x, "n'est pas divisible par", y
```

b) (3 pts) nombre premier

On rappelle qu'un nombre premier est un nombre entier positif qui n'a aucun autre diviseur entier que lui-même ou le nombre 1. Autrement dit, un nombre entier positif p est premier si et seulement si il n'existe aucun autre nombre entier positif q plus petit que lui et autre que 1, tel que p est multiple de q .

Ecrire un programme en Fortran 90 qui demande à l'utilisateur d'entrer un nombre entier au clavier et qui détermine s'il est, ou non, premier.

Pour répondre à la question, le fichier ~/exam/exo2/exo2b.f90 devra être complété.

```
1 PROGRAM exo2b
2
3 IMPLICIT NONE
4
5 LOGICAL :: est_premier
6 INTEGER :: n,i
7
8 WRITE (*,*) "Entrez un nombre entier s'il vous plait"
9 READ (*,*) n
10
11 est_premier = .TRUE.
12 DO i=2,n-1
13     IF (n/i == n/(i*1.0)) THEN
14         est_premier = .FALSE.
15     END IF
16 END DO
17
18 IF (est_premier) THEN
19     WRITE (*,*) "Le nombre",n," est premier."
20 ELSE
21     WRITE (*,*) "Le nombre",n," n'est pas premier."
22 END IF
23
24 END PROGRAM exo2b
```

c) (3 pts) nombres premiers jumeaux

Deux nombres premiers p et q sont dits "jumeaux" s'ils correspondent à deux nombres impairs consécutifs, c'est-à-dire si $q = p + 2$. Par exemple, (11, 13) est un couple de nombres premiers jumeaux.

Ecrire un programme en Fortran 90 qui demande à l'utilisateur d'entrer au clavier un nombre entier n et qui affiche à l'écran tous les couples de nombres premiers jumeaux inférieurs à n .

Pour répondre à la question, le fichier ~/exam/exo2/exo2c.f90 devra être complété.

```
1 PROGRAM exo2c
2
3 IMPLICIT NONE
4
5 LOGICAL :: n_premier, p_premier
6 INTEGER :: n, p, i, nmax
7
8 WRITE (*,*) "Entrez un nombre entier s'il vous plait"
9 READ (*,*) nmax
10
11 DO n=1,nmax
12
13     n_premier = .TRUE.
14     DO i=2,n-1
15         IF (n/i == n/(i*1.0)) THEN
16             n_premier = .FALSE.
17         END IF
```

```

18  END DO
19
20  IF (n_premier) THEN
21
22      p = n+2
23      p_premier = .TRUE.
24      DO i=2,p-1
25          IF (p/i == p/(i*1.0)) THEN
26              p_premier = .FALSE.
27          END IF
28      END DO
29      IF (p_premier) THEN
30          WRITE (*,*) "Le couple (" , n , " , " , p , ") est un couple de nombres premiers jumeaux"
31      END IF
32
33  END IF
34
35  END DO
36
37  END PROGRAM exo2c

```

Indiquer sur copie la liste des couples de nombres entiers jumeaux inférieurs à 100.

```

( 1 , 3 )
( 3 , 5 )
( 5 , 7 )
( 11 , 13 )
( 17 , 19 )
( 29 , 31 )
( 41 , 43 )
( 59 , 61 )
( 71 , 73 )

```

3) (12 pts) Un poisson dans l'eau

On considère une population de N poissons qui évoluent dans l'océan. Ceux-ci, de leur naissance jusqu'à leur mort, sont soumis à la menace de prédateurs qui cherchent à les attraper pour les consommer. On considère que chaque année, la probabilité de survie d'un poisson est égale à s , une constante comprise entre 0 et 1. Chaque année, chaque poisson effectue donc un tirage au sort pour savoir s'il est amené à survivre une année de plus ou à mourir entre les machoires du prédateur (probabilité s de survivre, probabilité $(1 - s)$ d'être mangé).

a) (2,5 pts) une vie de poisson

Ecrire un programme en Fortran 90 qui simule les tirages au sort de survie pour un poisson jusqu'à ce qu'il meurt. Le programme affichera alors la durée de vie du poisson.

On prendra, à partir de cette question, une valeur de probabilité de survie $s = 0.5$

Pour répondre à la question, le fichier `~/exam/exo3/exo3a.f90` devra être complété.

```

1  PROGRAM exo3a
2
3  USE sm05
4
5  IMPLICIT NONE
6
7  INTEGER :: age
8  REAL :: s ! proba de survie
9
10 s = 0.5
11
12 CALL sm05_graine(8)
13
14 age = 0

```

```

15 DO WHILE (sm05_tirage_uniforme()<s)
16     age = age + 1
17 END DO
18
19 WRITE (*,*) "Le poisson a vecu jusqu'a l'age de", age, "ans."
20
21 END PROGRAM exo3a

```

b) (1,5 pts) des vies de poissons

Ecrire un programme en Fortran 90 qui simule les tirages au sort de survie pour chacun des N poissons de la population, jusqu'à leur mort, et qui calcule et affiche la durée de vie moyenne dans la population.

On prendra, à partir de cette question, $N = 1000$

Pour répondre à la question, le fichier `~/exam/exo3/exo3b.f90` devra être complété.

```

1 PROGRAM exo3b
2
3 USE sm05
4
5 IMPLICIT NONE
6
7 INTEGER :: age, N, i, somme_age
8 REAL :: s ! proba de survie
9
10
11 s=0.5
12 N=1000
13
14 CALL sm05_graine(8)
15
16 somme_age = 0
17 DO i=1, N
18     age = 0
19     DO WHILE (sm05_tirage_uniforme()<s)
20         age = age + 1
21     END DO
22     somme_age = somme_age + age
23 END DO
24
25 WRITE (*,*) "L'esperance de vie moyenne dans la population est de", somme_age/(N*1.0), "ans."
26
27 END PROGRAM exo3b

```

c) (3 pts) le banc

Ecrire un programme en Fortran 90 qui simule le devenir des N individus de la population sur 10 ans et qui affiche, sous forme d'un graphique, l'évolution de la taille de la population (le nombre de poissons survivants de la population initiale) sur ces 20 ans.

Pour répondre à la question, le fichier `~/exam/exo3/exo3c.f90` devra être complété.

```

1 PROGRAM exo3c
2
3 USE sm05
4
5 IMPLICIT NONE
6
7 INTEGER :: N, i, an, an_fin, survivants
8 REAL :: s ! proba de survie
9 REAL, DIMENSION(0:10) :: popT, anneeT
10
11
12 s=0.5
13 N=1000

```

```

14 an_fin = 10
15
16 CALL sm05_graine(8)
17
18 anneeT(0) = 0
19 popT(0) = N
20
21 DO an = 1, an_fin
22     survivants = 0
23     DO i=1, N
24         IF (sm05_tirage_uniforme())<s) THEN
25             survivants = survivants + 1
26         END IF
27     END DO
28     N = survivants
29     anneeT(an) = an
30     popT(an) = N
31 END DO
32
33 CALL sm05_courbe(anneeT, popT, "annee", "nombre_d'individus", "evolution_du_banc")
34
35 READ (*,*)
36
37 END PROGRAM exo3c

```

d) (1,5 pts) un peu de sexe

Un poisson devient reproducteur à l'âge de 4 ans et produit alors 20 poissons juvéniles. On considère alors que l'effort de reproduction a épuisé le reproducteur et qu'il meurt aussitôt. On considère qu'au début de la simulation, tous les N poissons de la population ont 0 an. Ecrire un programme en Fortran 90 qui simule et détermine, sur les N poissons de la population, combien atteignent l'âge de reproduction (avant leur mort) et qui détermine alors le nombre total de juvéniles produits.

Pour répondre à la question, le fichier `~/exam/exo3/exo3d.f90` devra être complété.

```

1 PROGRAM exo3d
2
3 USE sm05
4
5 IMPLICIT NONE
6
7 INTEGER :: age, N, i, repro, fertilité
8 REAL :: s ! proba de survie
9
10
11 s=0.5
12 N=1000
13 repro = 0
14 fertilité = 20
15
16 CALL sm05_graine(5)
17
18 DO i=1, N
19
20     age = 0
21     DO WHILE (sm05_tirage_uniforme())<s .AND. age<4)
22         age = age + 1
23     END DO
24     IF (age==4) THEN
25         repro = repro + 1
26     END IF
27 END DO
28
29 WRITE (*,*) "Le_nombre_de_juveniles_produits_est_de_", repro*fertilité
30
31 END PROGRAM exo3d

```

e) (3,5 pts) plus de sexe

Les juvéniles produits lors de la première reproduction évoluent à leur tour et se reproduisent éventuellement eux aussi au bout de 4 ans. Ecrire un programme en fortran 90 qui simule l'évolution de la population, en prenant en compte ces reproductions, jusqu'à ce que la population s'éteigne ou au contraire persiste sur 100 générations. Dans le cas d'une extinction prématurée, le programme indiquera le nombre de générations pendant lesquelles la population a survécu.

Pour répondre à la question, le fichier `~/exam/exo3/exo3e.f90` devra être complété.

```
1 PROGRAM exo3e
2
3 USE sm05
4
5 IMPLICIT NONE
6
7 INTEGER :: age, N, i, repro, fertilité , gen_max, gen
8 REAL :: s ! proba de survie
9
10
11 s=0.5
12 N=1000
13 gen_max = 100
14 fertilité = 20 ! On peut passer a 16 pour acclereler le calcul
15
16 CALL sm05_graine(50)
17
18 gen = 0
19 DO WHILE (N>0 .AND. gen<gen_max)
20
21     repro = 0
22     DO i=1, N
23
24         age = 0
25         DO WHILE (sm05_tirage_uniforme())<s .AND. age<4)
26             age = age + 1
27         END DO
28         IF (age==4) THEN
29             repro = repro + 1
30         END IF
31     END DO
32
33     N = repro* fertilité
34     gen = gen + 1
35
36     WRITE (*,*) gen, N
37
38 END DO
39
40 IF (N>0) THEN
41     WRITE (*,*) "La population a survécu sur", gen_max,"generations (effectif final =",N,")"
42 ELSE
43     WRITE (*,*) "La population s'est éteinte à la generation",gen
44 END IF
45
46 END PROGRAM exo3e
```