

PHYS-4007/5007: Computational Physics

Fortran Programming in Linux, Part 2 Dealing with Strings in Fortran

1 Prep Work

Log into your account on the Linux side of the computers in Brown Hall 264. Open a terminal window, then check to make sure that you have a subdirectory called `fortran` in your login directory using

```
> ls
```

— the ‘>’ above represents the Linux system prompt (do not type this in). If you do not have such a subdirectory, make one using

```
> mkdir fortran
```

then change to this subdirectory using

```
> cd fortran
```

Now create a new file called ‘`mystrings.f`’ with the command

```
> emacs mystrings.f &
```

(remember that the ‘&’ sign put this job in background so that you can still access Linux in the terminal window).

2 Dealing with Strings in a Fortran Code

For this tutorial, you will be asked to enter and print out the place of your birth, the year you were born, and from this, figure out your age. You will be shown how to request text information, then process and print out this information. The code below will ask the user to input their birthplace city and state, followed by the year they were born. Then using a gfortran date and time function, the following is

printed to the terminal:

```
You were born in CITY, STATE in YEAR.  
You are AGE years old.
```

following the format

```
'You were born in ', String, String, ' in ', nnnn  
'You are ' nn ' years old.'
```

Since Fortran 77 cannot easily trim strings of trailing blanks, this program shows the student the quick way of printing the required output, and a more detailed way to print the output exactly as requested above.

In the **emacs** GUI (Graphic User Interface) window, type in the following lines (except for the first sequence of number lines):

```
123456789T123456...  
      PROGRAM MYSTRINGS  
C  
C Define variables.  
C  
      INTEGER IOUT, AGE, BYEAR, VALUES(8)  
      CHARACTER CITY*20, STATE*16, OUTPUT*40  
      CHARACTER DATE*8, TIME*10, ZONE*5  
C  
C Get information from the keyboard.  
C  
      PRINT *, 'Enter the name of your birth city:'  
      READ (*, 10) CITY  
      PRINT *, 'Enter the name of your birth state:'  
      READ (*, 10) STATE  
      PRINT *, 'Enter the year of your birth:'  
      READ (*, 20) BYEAR  
C  
C Get the current date from the computer and calculate your age.  
C VALUES(1) contains the current year.  
C  
      CALL DATE_AND_TIME(DATE, TIME, ZONE, VALUES)  
      AGE = VALUES(1) - BYEAR  
C  
C Display the output, first in raw form.  
C  
      WRITE(*, 30) CITY, STATE, BYEAR  
      WRITE(*, 50) AGE
```

```

C
C Now print the required output in the "pretty" format requested in the
C description above. The logic below assumes that the city name is only
C one word.
C
      IOUT = 1
      DO 100 I = 1, 20
        IF (CITY(I:I) .NE. ' ') THEN
          OUTPUT(IOUT:IOUT) = CITY(I:I)
          IOUT = IOUT + 1
        ENDIF
      100 CONTINUE
      OUTPUT(IOUT:IOUT+1) = ', '
      IOUT = IOUT + 2
      DO 110 I = 1, 16
        IF (STATE(I:I) .NE. ' ') THEN
          OUTPUT(IOUT:IOUT) = STATE(I:I)
          IOUT = IOUT + 1
        ENDIF
      110 CONTINUE
C
C Subtract IOUT by 1 to get back to the actual number of characters
C in OUTPUT.
C
      IOUT = IOUT - 1
C
      PRINT *, ' '
      WRITE(*, 40) OUTPUT(1:IOUT), BYEAR
      WRITE(*, 50) AGE
C
C Format statements.
C
      10  FORMAT(A)
      20  FORMAT(I4)
      30  FORMAT(' You were born in ', A20, ', ', A16, ' in ', I4, ' .')
      40  FORMAT(' You were born in ', A, 1X, 'in ', I4, ' .')
      50  FORMAT(' You are ', I2, ' years old.')
C
      STOP
      END

```

Note that **Fortran** has the ability to add strings together too in order to make a new string using the `//` operator. For instance,

```
OUTPUT = CITY // ', ' // STATE
```

This `OUTPUT` string variable would then contain the 16 character string `STATE` attached to the 2 character string `', '` attached to the 20 character string `CITY`. Feel free to play around with this operator in the code above.

To compile this code from the **terminal** window, issue the following from the **Linux** command prompt (`>`):

```
> gfortran -o mystrings.exe mystrings.f
```

then to run the code from the terminal window:

```
> ./mystrings.exe
```

then answer the questions that the program requests. The first 2 lines of output shows you what the the output would look like if you had not tried to format the output the way it was originally requested, while the second 2 lines shows you the output as requested.