Assignment One : Part I

MT4112

Deadline Sunday 23^{rd} Oct 11:00pm

All work **must** be done in the sub-directory '`assign1`' off your home directory. When the assignment deadline has passed you will not be able to write to this directory anymore but you can still read what you have done. The work will then be copied from the '`assign1`' directory and then marked.

Question One

Type the answers to question one into a file called ' `quest1ans.txt`' inside your '`assign1`' directory using 'nano'.

- [**a**] Write out the expressions below and for each of the operators, using the numbers $1 \rightarrow n$ (where n is the number of operators) indicate, above the operator, the order in which the operations will be performed (as on page one of handout three).

```
a + b - (c * d) / f + 5.0 ** e

a + f * ( b - c ) / ( d - 3 ) - e * f

b * ( c / d ) - ( f + a ) / 3.0 * e

(( f - e ) * 4 ** c ) * a / d - b

a ** b ** c + d / ( e + f ) * 2
```

- [**b**] Consider following code

```
PROGRAM q1partb

  IMPLICIT NONE

  INTEGER :: a=7,b=2,c=9
  REAL    :: d=5,e=2,f=3

  PRINT*,b-d*a-c-f**2+b/c

END PROGRAM q1partb
```

1. What do you think the answer printed to the screen will be?
2. What do you think the 'data type' of the answer will be?
3. Qualify your answers to 1 and 2 above by creating a table of three columns. The first column holds the operators (top $\rightarrow$ bottom) in the order they would be executed in the expression. The second column holds the numeric result of each operation corresponding to the operator

in the same row of the first column. The third column holds the data type of the numeric result for each operation.

Question Two

You need to write a code that implements an algorithm to reduce a fraction to its lowest terms. This means that there is no number, other than "one" that can be divided exactly into both the numerator and the denominator. To do this you will need to work out the the greatest denominator common to both the numerator and the denominator of the fraction, this is often referred to as the greatest common denominator (gcd). Your code will need to work for both proper and improper fractions.

- Write your source code using nano into a file called '`fraction.f90`' and compile the source code into an executable file called '`fraction`'.
- Declare the appropriate required variables including two '`INTEGER`' types one for the numerator and one for the denominator.
- Read in the numerator and the denominator of the fraction to be reduced using the fortran '`READ`' command.
- You should make use of a **single** '`DO`' loop in your code so you can calculate the greatest common denominator (gcd).
- Take care over how you calculate the limits of the '`DO`' loop. This will have a significant influence on the efficiency of your code. HINT: There is no point testing the numerator or denominator for factors that could not possibly be factors due to their magnitude.
- You will also need to use '`IF`' constructs in your code, in particular one inside the '`DO`' loop to test for the existence of a common factor.
- Each time you find a new common factor your code should print it out to the screen. See the example output below.
- You will find the fortran function '`MOD`' useful. This function takes two arguments of the same type. It divides the second argument into the first argument and then calculates and returns the remainder. So for example '`PRINT*,MOD(14,3)`' would print '`2`' to the screen as '`3`' goes into '`14`' four times with a remainder of '`2`'.
- You may find the fortran '`MIN`' function useful. This function accepts a comma separated list of values of the same type and returns the minimum value in the list. So for example given '`b`' is of '`INTEGER`' type. Then

  ```
  b=2
  PRINT*,MIN(4,9,6,8,b)
  ```

  would print '`2`' to the screen.
- You may well find the fortran rounding function '`FLOOR`' useful. This function will return the largest integer not exceeding the magnitude of its '`REAL`' argument. For example '`FLOOR(3.9)`' would return '`3`' and '`FLOOR(17/2.0)`' would return '`8`'.
- Make sure you use the '`IMPLICIT NONE`' statement, comment your code adequately and structure any '`DO`' loops and '`IF`' structures appropriately.

Please use your code to reduce the following fractions.

$$\frac{12}{72}, \quad \frac{17}{102}, \quad \frac{256}{1024}, \quad \frac{98}{24} \tag{1}$$

As mentioned above write your code in the directory '`assign1`' in a file called '`fraction.f90`'. Redirect your code's output, for each of the above fractions, from the screen into the same file called '`quest2ans.txt`'. Please ensure the answers appear in the same order as the fractions do in the list. The output from your code should appear as below for the first fraction $\frac{12}{72}$.

```
 Please read in the numerator
 Please read in the denominator

 Numerator input     =          12
 Denominator input   =          72

 Found a new factor :           2
 Found a new factor :           3
 Found a new factor :           4
 Found a new factor :           6
 Found a new factor :          12

 Numerator   =           1
 Denominator =           6
```