Practical Exercise 1

Question 1: The Hello World Program
Write a Fortran 95 program to write out `Hello World` on the screen.

Question 2: Real Formatting
Write a program which uses the expression `4.0*atan2(1.0,1.0)` to evaluate π and store it in a variable. Write out this value 9 times using edit descriptors of the form `E12.d, F12.d, G12.d` with `d` taking the values 2, 4 and 6.

Question 3: Some Division One Results
A particular number can be expressed as the sum of several integers, and the sum of the reciprocals of these integers is, perhaps, surprising. Write a program to calculate the values of the two following expressions and write a short text and the results:

```
2 + 6 + 8 + 10 + 12 + 40
```

$$\frac{1}{2}+\frac{1}{6}+\frac{1}{8}+\frac{1}{10}+\frac{1}{12}+\frac{1}{40}$$

Hint: some constants are better as type `INTEGER` but some must be type `REAL`.

Now write similar results using the set of numbers `{2,3,10,24,40}`

Question 4: Area of a Circle
Write a simple program to read in the radius and calculate the area of the corresponding circle and volume of the sphere. Demonstrate correctness by calculating the area and volume using radii of 2, 5, 10.

Area of a circle:

$$area = \pi r^2$$

Volume of a sphere:

$$volume = \frac{4\pi r^3}{3}$$

Hint: use the value 3.14159 for π.

```
PROGRAM Area_and_Vol
!...Add specification part
  WRITE(*,"(A)") "Type in the radius: "
  READ*, radius
  !...Add code to calculate area and volume
  WRITE(*,"(A26,F5.1,A4,F6.1)") &
     "Area of circle with radius ",&
     radius, " is ", area
  WRITE(*,"(A28,F5.1,A4,F6.1)") &
```

```
        "Volume of sphere with radius ",&
        radius, " is ", volume
    END PROGRAM Area_and_Vol
```

Question 5: Filed values
Write a program to open the file named `statsa` which has been provided: `statsa` contains several values, each on a separate line (or record). Read the first value which is an integer, and is in a field of width 5. Then read the second value which is of type real, in a field of width 5 with two digits after the decimal point. Write these two values within a line of explanatory text to the screen.

Now generalize your program by reading the name of the file into a character variable and using this character variable in the `OPEN` statement.

Practical Exercise 2

Question 1: Parity
Write a program to read several numbers, positive or negative, one at a time and for each to write out a line giving the number just read and a description of it as an odd or even number. Stop if the number read in is zero.

Question 2: A Triangle Program
Write a program to accept three (`INTEGER`) lengths and report back on whether these lengths could define an equilateral, isosceles or scalene triangle (3, 2 or 0 equal length sides) or whether they cannot form a triangle.

Demonstrate that the program works by classifying the following:

1. (3, 3, 3)
2. (3, 3, 4)
3. (3, 4, 5)
4. (3, 3, 7)

Hint: If three lengths form a triangle then 2 times the longest side must be less than the sum of all three sides. In Fortran 95 terms, the following must be true:
`(2*MAX(side1,side2,side3) < side1+side2+side3)`

Question 3: The Ludolphian Number
Write a program which uses 6 variables of type real; `a`, `b`, `c`, `d`, `e`, `f` (or any other names you choose). Set initial values as follows, remembering to match the type of constant to the type of variable:

$$a = 1, \quad b = \frac{1}{\sqrt{2}}, \quad c = \frac{1}{4}, \quad d = 1$$

Code these 7 lines as Fortran 95 statements (with constants of the correct type) within a loop which is to be obeyed 4 times:

$e = a$

$a = \frac{(a+b)}{2}$

$b = \sqrt{b \times e}$

$c = c - d \times (a-e)^2$

$d = 2d$

$f = \frac{(a+b)^2}{4c}$

output f

This algorithm was developed by Tamura and Kanada.

Question 4: Odd Numbers

Write a program which:

1. Asks how many odd numbers you want to use.
2. Reads in the number of odd numbers to use(16 would be sufficient to test your program).
3. Sums this many odd numbers, starting from 1 (Do not use the formula for the sum of an arithmetic progression!)
   As each number is added in, write out a count of how many odd numbers have been added in and what the sum is. So the first line will simply be:
   1 1

Question 5: Simple Sequences (symmetric, unitary, descending)

For each of these sequences set an initial value and use a DO-loop.

a) Write a program to evaluate and write out each of the terms in this sequence:

1 x 1
11 x 11
111 x 111
:
11111 x 11111

Now evaluate and write out the next term in this sequence. Anything strange?

b) Write a program to evaluate and write out each of the terms in this sequence:

0 x 9 + 1
1 x 9 + 2
12 x 9 + 3
123 x 9 + 4
:
12345678 x 9 + 9

c) Write a program to evaluate and write out each of the terms in this sequence:

1 x 8 + 1

12 x 8 + 2
123 x 8 + 3
:
123456789 x 8 + 9

Question 6: Mathematical Magic
If you take a positive integer, halve it if it is even or triple it and add one if it is odd, and repeat, then the number will eventually become one. This is known as the Syracuse algorithm.

Set up a loop containing a statement to read in a number (input terminated by zero) and a loop to write out the sequence obtained from each input. When the number written out is 1 then execution should terminate with an appropriate message.
Demonstrate that your program works by outputting the sequences generated by the following sets of numbers:

a) 7
b) 106, 46, 3, 0

Question 7: Rooting
Write a program which uses 2 variables of type real; `a`, `x` (or any other names you choose). Issue prompts and read in initial values as follows:

`a` is the number whose square root we are finding,
`x` is an estimate of the root.

Code these 2 lines as Fortran 95 statements within a loop which is to be obeyed several times, say 6 times:

```
x = (x + a/x)/2
```
output `x`

The algorithm used here is the Newton-Raphson one.
You might be interested to compare your result with that given by the intrinsic function `sqrt(a)`

Question 8: Coins
Assume you have coins with face values 50, 20, 10, 5, 2 and 1. Write a program which reads in the price of some item which is not greater than 100 and finds the fewest number of coins whose sum equals this price. Write out how many of each value coin is used: stop if the original price is 0.

Question 9: Vowel, Consonant or Other
Using a `SELECT CASE` block write a program that reads in any number of characters, one at a time, and for each character writes out whether it is a vowel, a consonant or neither: read in the '@' character to terminate the input.

Question 10: Decimal to Roman Numerals Conversion
Using a `SELECT CASE` block and integer division write a program that reads in a decimal number between 0 and 999 and writes out the equivalent in Roman Numerals.
Demonstrate that your program works with the numbers:
1. 888
2. 0
3. 222
4. 536

The output should contain no embedded spaces.

0					
1	i	1.	x	1..	c
2	ii	2.	xx	2..	cc
3	iii	3.	xxx	3..	ccc
4	iv	4.	xl	4..	cd
5	v	5.	l	5..	d
6	vi	6.	lx	6..	dc
7	vii	7.	lxx	7..	dcc
8	viii	8.	lxxx	8..	dccc
9	ix	9.	xc	9..	cm

Hint: Use a `CHARACTER` string (or `CHARACTER` strings) to store the number before output. The `longest' number is 888, dccclxxxviii (12 characters).

Practical Exercise 3

Question 1: Rank, Bounds etc.
Give the rank, bounds, shape and size of the arrays defined as follows:

```
REAL, DIMENSION(1:10) :: ONE
REAL, DIMENSION(2,0:2) :: TWO
INTEGER, DIMENSION(-1:1,3,2) :: THREE
REAL, DIMENSION(0:1,3) :: FOUR
```

Write down the array element order of each array.

Which two of the arrays are conformable?

Question 2: Array Sections
Declare an array which would be suitable for representing a chess board. Write a program to set all the white squares to zero and the black squares to one. (A chess board is 8×8 with alternate black and white squares.) Use formatted output to display your chess board on the screen.

Question 3: Array Constructor
Euler noted that a sequence of 40 prime numbers p starting at 41 can be found from the formula:

$$p = 41 + x + x^2, \quad \text{for } 0 \le x \le 39$$

Write a program using an array constructor to store this sequence of 40 primes in an array, putting the first prime in element 0 or 1. Use formatted write to output the sequence on your screen, with at most 5 primes on each row.

Question 4: Fibonacci Numbers
The Fibonacci numbers are defined as follows:

$$u_0 = 0;\ u_1 = 1;\ u_n = u_{n-2} + u_{n-1} \text{ for } n \ge 2$$

Write a program to generate and store in an array the Fibonacci numbers up to and including u_{24}.

The sum of the first n numbers is $u_{n+2} - 1$. Use the intrinsic function SUM on an array section to find the sum of the numbers u_0 to u_{22}. Compare this result with the value of $u_{24} - 1$.

The sum of the first n numbers with odd indices is:

$$u_1 + u_3 + u_5 + \ldots + u_{2n-1} = u_{2n}.$$

Use the intrinsic function SUM on an array section to find the sum of the numbers with odd indices up to u_{23}. Compare this result with the value of u_{24}.
The sum of the first n numbers with even indices is:

$$u_2 + u_4 + u_6 + \ldots + u_{2n} = u_{2n+1} - 1.$$

Use the intrinsic function SUM on an array section to find the sum of the numbers with even indices up to u_{22}. Compare this result with the value of $u_{23} - 1$.

Question 5: Magic Squares
A magic square is a set of numbers arranged in a square array so that the sum of the numbers in each row, the sum of the numbers in each column and the sum of the numbers along each diagonal are all equal. This sum is known as the magic number of this particular magic square.
Write a program to create two 3 x 3 arrays holding these magic squares:

4	9	2
3	5	7
8	1	6

9	2	7
4	6	8
5	10	3

- For each magic square write a line of text as a heading and then the magic square.
- Add the two magic squares together and save the result in a third array: write a heading and then this magic square.
- Check that this is a new magic square by comparing the sums across the first row, down the last column and along the leading diagonal.

Question 6: Symmetry
Write a program to work with the first magic square of Question 5.

- Write the square's magic number (the row, column or diagonal sum). You can check your answer because for an n x n magic square consisting of any arrangement of the integers 1 to n^2 the formula is $(n^3 + n)/2$
- Use the intrinsic function `TRANSPOSE` to save the transpose of the magic square in a new 3 x 3 array.
- Add the magic square to its transpose and save the result in a new array: this should be a symmetric matrix. Check that the bottom left and top right elements are equal: write out the symmetric matrix.

Question 7: More Magic
Modify the Mathematical Magic program which you wrote for Exercise 2, Question 6 to save the sequences generated in an array. Write out each sequence and find the largest value in each of these sequences and the position in the sequence at which it occurs.

Question 8: MATMUL Intrinsic
For the declarations
```
REAL, DIMENSION(8,8) :: A, B, C
```
what is the difference between `C=MATMUL(A,B)` and `C=A*B`?

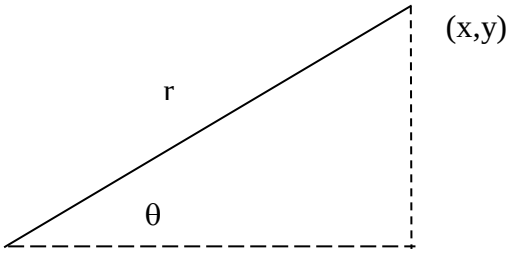
Question 9: More Filed values
Modify the Filed values program which you wrote for Exercise 1, Question 5 to declare an allocatable rank one array of type real. Use the integer value which is read in from the file `statsa` as the upper bound for the array when it is allocated (and make the lower bound 1). Then fill the array with real values read from the file. (All values are in fields of width 5 with two digits after the decimal point.) Write out these real values, with 5 on each line. Deallocate the array.

Practical Exercise 4

Question 1: Point on a circle
Write a program to read in a vector defined by a length, r, and an angle, θ, in degrees which writes out the corresponding (x, y) co-ordinates. Recall that arguments to trigonometric functions are in radians.

Demonstrate correctness by finding the (x, y) co-ordinates for the following vectors:

1. r = 2, θ = 60°
2. r = 3, θ = 120°
3. r = 5, θ = 240°
4. r = 8, θ = 300°
5. r = 13, θ = 450°



Hint: remember that $\sin\theta = \frac{y}{r}$ and $\cos\theta = \frac{x}{r}$ and 180 degrees = π radians

Question 2: Simple example of a Subroutine
Write a main program and an internal subroutine that returns, as its first argument, the sum of two real numbers.

Question 3: Simple example of a Function
Write a main program and an internal function that returns the sum of two real numbers supplied as arguments.

Question 4: Switch or Stick
Write a main program and an internal subroutine with two arguments that returns, as its first argument, the smaller of two real numbers and as its second argument, the other number.

Question 5: Standard Deviation
Write a program which contains an internal function that returns the standard deviation from the mean of an array of real values. Note that if the mean of a sequence of values (x_i , i = 1, n) is denoted by m then the standard deviation, s, is defined as:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - m)^2}{(n-1)}}$$

[Hint: In Fortran 95 `SUM(X)` is the sum of the elements of `X`.]
To demonstrate correctness write out the standard deviation of the following numbers (10 of them):

5.0 3.0 17.0 -7.56 78.1 99.99 0.8 11.7 33.8 29.6

and also the following 14:

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0

The files `statsa` and `statsb` contain these two sets of real values preceded by the relevant count (see Exercise 3, Question 9).

Question 6: Save Attribute
Write a skeleton procedure that records how many times it has been called.

Practical Exercise 5

Question 1: Encapsulation

Define a module called `Simple_Stats` which contains encapsulated functions for calculating the mean and standard deviation of an arbitrary length `REAL` vector. The functions should have the following interfaces:

```
REAL FUNCTION mean(vec)
     REAL, INTENT(IN), DIMENSION(:) :: vec
END FUNCTION mean

REAL FUNCTION Std_Dev(vec)
     REAL, INTENT(IN), DIMENSION(:) :: vec
END FUNCTION Std_Dev
```

[Hint: In Fortran 95, `SIZE(X)` gives the number of elements in the array `X`.]
You may like to utilise your earlier code as a basis for this exercise.
Add some more code in the module, which records how many times each statistical function is called during the lifetime of a program. Record these numbers in the variables: `mean_use` and `std_dev_use`.
Demonstrate the use of this module in a test program; in one execution of the program give the mean and standard deviation of the following sequences of 10 numbers:

5.0 3.0 17.0 -7.56 78.1 99.99 0.8 11.7 33.8 29.6

and then the following 14:

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0

Write out the values of `mean_use` and `std_dev_use` for this run of the program.

Question 2: Binary Cut

Write a module containing a function which returns the position of a particular number in an array of sorted integers. The function should employ the so-called ``binary cut'' method. This method proceeds by determining in which half the number is and then concentrating on that half. It is easily implemented by using two indices to point at the low and high positions of the current area of interest. It is assumed that if there is more than one occurrence of the number then the one with the higher index will be chosen. This method is very efficient for very large arrays.

Algorithm:

- Let `i` and `j` be the indices (subscripts) to (array) `x` of the low and high marks.
- Initially set `i = 1` and `j = n` (the number in the list)
- Assume `k` is the number we are trying to find
- `DO`
- `IF(i ≥ j)EXIT`
- determine the half way point $ihalf = \frac{i+j}{2}$
- `IF k` is above `x(ihalf)` put `i = ihalf + 1`
- Otherwise put `j = ihalf`
- `END DO`
- `j` will now point at the number `k`

Question 3: Spheres Apart

Write a program to look at the relationship between all pairs of an arbitrary number of spheres defined in 3-dimensional space. Read in the number of spheres being used and read the coordinates of the centres and the lengths of the radii of these spheres into an allocatable array of a defined type variable. For spheres s_m and s_n the separation of their centres is given by the formula:

$$\sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2}$$

If the centre of one sphere lies within the other then output a line stating this fact. Otherwise are the surfaces of the two spheres touching, intersecting or separate? You could try your program on spheres with these centres and radii:
(3.0,4.0,5.0), 3.0 (10.0,4.0,5.0), 4.0 (3.0,-3.0,5.0), 5.0 (3.0,4.0.8.0), 6.0

Question 4: Real Portability

Take a copy of the program you wrote in Question 3 of Exercise 2 to find the Ludolphian number. Replace the statement of the form:

```
REAL :: a, b, c, d, e, f
```

by the statements of the form:

```
INTEGER, PARAMETER :: k = SELECTED_REAL_KIND(P=15,R=31)
REAL(KIND=k)       :: a, b, c, d, e, f
```

Add a statement to check that `k > 0`, and change the kind of the constants to `k`, for example `1.0_k` Output the results with 12 decimal digits.

Run this program and compare the results with those you got earlier.

Question 5: Integer Portability

Take a copy of the program you wrote in Question 5(a) of Exercise 2 to find the first 5 terms of a sequence. Extend the range of those integers necessary to find the 6th term of this sequence.