In-class Exercise 1 (Open book, 25 minutes)

Student ID_________________________Name (Print)__________________________

Loop: Chapman 4.1
Function: Chapman 7.4

1. Write a Fortran 95/2003 function for the following math function:

$$ f(x, y, p) = \left( x^p + y^p \right)^{\frac{1}{p}} $$

The name of the Fortran function is pNorm, the input arguments are x, y, and p, the data type of x and y is real, the data type of p is integer, the output of the function is of real type.

!start your function here

```fortran
real function pNorm( x, y, p )
   implicit none

   !start your input argument declaration here
   real, intent(in)::x,y
   integer, intent(in)::p

   !start your formula translation here
   pNorm = (x**p + y**p)**(1.0/p)

   ! Notice (1.0/p), 1/p will yield 0 for p>1 due to integer truncation

!close your function here
end function
```

2. Write a Fortran 95/2003 program that can generate a table which display the pNorm result for a set of 2D points. The point set is defined as follows:

```
    y
      a+4d--------.-------- x
           .
           .
     a+d-------- x
           .
           .
     a . . . .
   a  a+d  . . .    a+4d
```

The points are along the line of y-x=0, the first point is at \((a, a)\), then the following points are all equally spaced with a identical increments in the x and y axis of \(d\). The user inputs of the program are the values of \(a\) and \(d\) in real type, the number of points \(n\) in integer type, and the degree of the norm \(p\) in integer type. The output display table should have the following information:

- point index, \(x\), \(y\), the corresponding pNorm value

You can name your program `pNormTable`. !start your program here

```fortran
program pNormTable

implicit none

!declare your variables here
real :: a=0., d=0.
integer :: n=0, p=1
real :: x=0.
integer :: i=0
!declare pNorm function here
real :: pNorm
write(*,*), "enter a, d, n, and p:"
read(*,*) a,d,n,p
!start a counting loop here
do i=0, n-1

!start the statement that your want to repeat in the loop
x = a + i*d
write(*,*) i+1, x, x, pNorm(x, x, p)

!close your loop here
end do

!close your program here
end program
```
3. Please implement the following algorithm illustrated in the flowchart into a Fortran 95/2003 function.

Before you start, first determine your function name using the output information in the flowchart:

riemannSum

Then list all the input argument:

a, b, n

The type for input a and b and function output is real, the type for input n is integer.

Now list all the local variables appeared in the flowchart:

dx, x

Now, search if there is any function calls in the flowchart, assume all input and output of any functions are all in real type:

foo

Now identify the best loop construct, do loop, do-while loop, or counting

do-while

!start your function here
real function riemannSum( a, b, n )
    implicit none
    !start input argument declaration here
    real, intent(in)::a,b
    integer, intent(in)::n

    !start local variable declaration here
    real::dx=0,x=0

    !start function declaration here
    real::foo
!now you're done with the first block “start” in the flowchart, !continue the implementation of the following two blocks
dx = (b-a)/n
x = a+dx/2
riemannSum = 0

!now start your loop here, remember which loop construct you decided
do while ( x < b )
    riemannSum = riemannSum + foo(x) * dx
    x = x + dx

!close your loop here
end do

!close your function here
end function