Homework 1, due Wednesday, 8/26

0. **Warmup (not to turn in):** Test your editor and compiler by running a simple Fortran program:

```
program test
   implicit none

   integer :: i

   do i=1,5
       write(*,*) 'Hello ',i
   enddo

end program test
```

   - Using an editor type in the program. Save it as test.f90
   - to compile it:

     ```
     gfortran test.f90
     ```

   - to run it:

     ```
     ./a.exe
     ```

   - To save the output of your programs to a file run it like this:

     ```
     ./a.exe > output.txt
     ```

1. **Over/Underflows Exercise:** 1.5.2 page 24-25, all parts

   There is a sample program you can start with (overflow.f95). Download it from MyCourses—many of the programs on the CDROM have errors that I have fixed. Further suggestions:

   - you may need to increase N if the initial loop upper bound does not lead to underflow and overflow
   - the sample program divides each number by 2. Try multiplying or dividing by a number smaller than 2. Does this change the answer? Why?
   - you can use a separate program for parts 2(a), 2(b), and 2(c) or put them all in one program

2. **Machine Precision 1.5.4:** page 27. Do this for both single and double precision (real (kind=4) and real (kind=8)).

   Sample program is limit.f95

3. **Summation:** Suppose you need to sum the following series:

$$\sum_{n=0}^{\infty} \frac{1}{(2n+1)^2} = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \cdots = \frac{\pi^2}{8} \approx 1.233700$$

Try summing this series with a finite upper index n_{max}, increasing n_{max} to see how well the sum converges. Perform the sum in two different ways: i) with n increasing, and ii) with n decreasing. Calculate the relative error of the two sums. Compare the accuracy of the two ways as n_{max} increases- what is different? Use single precision `real(kind=4)` for this program to make the errors more obvious.

4. **PH6433 only: algorithms for e^{-x}**

   The simplest way of calculating e^{-x} is to simply write out the Taylor series and perform a direct sum:

$$e^{-x} = \sum_{n=0}^{\infty} (-1)^n \frac{x^n}{n!}$$

   This does not work well because the factorial quickly overflows. A better method is to use a recursion relation:

$$e^{-x} = \sum_{n=0}^{\infty} s_n \qquad s_n = -s_{n-1}\frac{x}{n}$$

   (define $s_0 = 1$ then determine s_1 from s_0, etc). There are some problems here too however.

   Write a program to calculate e^{-x} using this recursion relation:

   (a) To determine how many terms to sum, exit the loop that calculates the sum when $|s_n| < \epsilon$, where ϵ is a small number:

```
real(kind=8),parameter : eps=1.0e-10_8
x=10.0d0
n=0
term=1.0_8
sum=term
do
   n=n+1
   term= -term*x/n
   if (abs(term)<eps) exit
   sum=sum+term
enddo
```

   Test this for increasing x ($x = 0, 10, 20, 30, \ldots, 100$). What happens and why?

   (b) The fix is quite simple: First calculate

$$e^{x} = \sum_{n=0}^{\infty} s_n$$

   then take the inverse:

$$e^{-x} = \frac{1}{e^x}$$

   Show that this method gives accurate results.

I will distribute information on how to format your homework and turn it in separately.