Numerical Analysis and Computational Physics PHN 311, 2023
Dept. of Physics, IIT Roorkee

Assignment: Errors and precision of machine numbers

1. Consider the 32-bit single-precision floating-point number A

	s	*e*	*f*
Bit position	31	30 23	22 0
Value	0	0000 1110	1010 0000 0000 0000 0000 000

a) What are the (binary) values for the sign s, the exponent e, and the fractional mantissa f ?
b) Determine decimal values for the biased exponent e and the true exponent p.
c) Show that the mantissa of A equals 1.625 000.
d) Determine the full value of A.

2. Underflow and overflow limits

Write a program in any computer language, that determines the underflow and overflow limits within a factor of 2 for that language and the particular computer.

a) Check where under- and overflow occur for double-precision floating-point numbers (floats). Give your answer in decimal.
b) Check where under- and overflow occur for single-precision floating-point numbers.
c) Check where under- and overflow occur for integers, i.e. smallest and largest integers.
Note: There is no exponent stored for integers, so the smallest integer corresponds to the most negative one. To determine the largest and smallest integers, you must observe your program's output as you explicitly pass through the limits. You accomplish this by continually adding and subtracting 1. (Because integer arithmetic uses two's complement arithmetic, you should expect some surprises.)

Hint: Here is a sample pseudo-code:

```
under = 1.
over = 1.
begin do N times
     under = under/2.
     over = over * 2.
     write out: loop number, under, over
end do
```

3. Machine precision
Write a program to determine the machine precision ε_M of the computer you are using.
Hint: Starting with a value of 1.0, x is divided repeatedly by 2 until numerical addition of 1 and x = 2^{-M} gives 1.

a) Determine experimentally the precision of double-precision floats.
b) Determine experimentally the precision of complex numbers.

Note: To print out a number in the decimal format, the computer must convert from its internal binary representation. This not only takes time, but unless the number is an exact power of 2, leads to a loss of precision.

4. Use the diagnostic tool machar (machine characteristics) provided by "Numerical Recipe by Press, Teukolsky, Vetterling and Flannery" Page 889-894
https://www.cec.uchile.cl/cinetica/pcordero/MC_libros/NumericalRecipesinC.pdf

A c++ version of the code is here:
https://www.astro.umd.edu/~ricotti/NEWWEB/teaching/ASTR415/InClassExamples/NR3/index.htm

and compare with your resuts.

5. Truncation Error
This computer experiment approximates the cosine function by a truncated Taylor series

$$\cos(x) \approx \text{mycos}(x, n_{\text{max}}) = \sum_{n=0}^{n_{\text{max}}} (-)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \cdots \quad (1.65)$$

in the interval −π/2 < x < π/2. Compare the function mycos(x, n max) numerically to the intrinsic cosine function.