Numerical Analysis and Programming

Lab Worksheet #4

1. The following function approximately determines the *machine epsilon* (ϵ) for a given type (defaults to `float`), using the definition that ϵ is the smallest positive number such that $1 + \epsilon \neq 1$.

```
def machineEpsilon(func=float):
    machine_epsilon = func(1)
    while func(1)+func(machine_epsilon) != func(1):
        machine_epsilon_last = machine_epsilon
        machine_epsilon = func(machine_epsilon) / func(2)
    return machine_epsilon_last
```

   Understand what the function does, and determine ϵ for `int`, `float`, and `complex`. How is `float` in Python represented, single or double precision?

2. To convert a float pointing number between the machine representation in hexadecimals and the decimal representation using Python, we will use the `struct` module in Python. In this exercise, we will use two functions in the module, `pack` and `unpack`, to perform the task (`http://docs.python.org/2/library/struct.html`). The following code perform conversion of a hexadecimal representation of bit pattern `BF4F9680` to an unsigned integer,

```
import struct
output=struct.unpack("I", struct.pack("I",0xBF4F9680))[0]
print output
```

   What if you change the format string `"I"` in `unpack` to `"i"`? How about to `"f"`?

   Use the program to

   (a) Determine the decimal numbers that have the following machine representations: a. $[3F27E520]_{16}$, b. $[3BCDCA00]_{16}$, c. $[BF4F9680]_{16}$.

   (b) Determine the machine representation in hexadecimals in IEEE single precision for the following decimal numbers: a. 2^{30}, b. 64.015625, c. 8×2^{24}. Use the bulid-in functions `hex()` to convert an unsigned integer to hexadecimals, and `bin()` to binaries.