IC152: Assignment 1
Bitwise Operations and Mathematical Calculations

Important: This is just a practice assignment. There are 5 marks for attendance. TAs will conduct a mock viva for you to understand how vivas will happen in subsequent labs for evaluation.

Guidelines:

- One and only one of the members (first one who filled the form) per group should submit the pdf on LMS with solutions in the form of pdf by 22 Aug'24 9 p.m.
    - Submit one solution per question representing the whole group. Learnings you can put from different students in the group.
- If you are using chatGPT or any Large Language Model (LLM), mention about it clearly which part of your pdf is created using LLM, and which LLM you have used.
    - **note that LLMs also hallucinate or make incorrect predictions. If the answer by LLM is incorrect, it is your responsibility to correct it.**
- In any case you have to mention the learnings of your group after each question.
- You can discuss assignment questions/solutions within your group, discussing outside your group or sharing solutions with another group is **strictly not allowed**.
    - If any such case is found, the student will lose a grade in the course.
    - If a student is found twice sharing solutions, the student will get F grade and the case will be reported to the academic office.

Firstly, familiarize yourself with the Spyder environment using slides here.

This assignment will introduce us to the Python programming language. We will perform some bit wise operations and basic mathematical calculations using Python.

While the basic operations such as addition, subtraction, division and multiplication are readily available in the Python environment, other more advanced functions such as square-root, logarithm, etc. are provided by the math library.

Such libraries may be included in your program by adding a statement such as “import math” at the beginning of your program. Thereafter, you may use the square-root function by calling the function math.sqrt(x) which will return the square-root of the number x. Likewise, math.log(x, y) returns the logarithm of x to the base y. Math.factorial(3) gives the result of 3!, i.e. 3 X 2 X 1. The questions follow. Open Spyder Console or IDLE Shell in interactive mode and try to implement the following questions:

1. When you will learn advanced topics, you will see a lot of them use powers of two to optimize the programs. E.g. While training deep neural networks, a batch size in powers of two is generally used (batch size means number of samples a model trains on in parallel). In graphics, game designers use matrices of size four to transform objects from one frame of reference to another, although the objects and graphics environments have three dimensions.
   The bitwise operations have a relation with powers of two as we will see or have seen in the lectures, and will see again in the following questions:
    a. On your Spyder Console or IDLE Shell, print OR of two numbers: 2, 3 (operator for OR is |). What do you observe? Use bin() function to justify your answer.
    b. Assign a natural number to a variable with name n. Find if n is odd or not using & operator. The command should print 1 if the number is odd, else 0.
    c. Divide 24 by 2 using bit wise operations (Hint: use right shift operator: >>)

2.
    a. Print the result of 27/8 on your Spyder Console or IDLE Shell
    b. Print the result of 27//8 on your Spyder Console or IDLE Shell
    c. Divide 27 by 8 using bit wise operations and reason why you get same answer as previous question (i.e. 2-b)
    d. Using bit wise operations, print 2^{10} and 2^{20}. Verify your answers by printing 2**10 and 2**20.
    e. What will be the value of binary number 1111 in the decimal number system? Write a formula for converting the value of n bit binary number with 1 at all locations to the decimal number system. Verify your answer using the bin() function.

3.
    a. Print numbers -5 and 7 on the same line
    b. Do you know that 1729 is known as the Hardy-Ramanujan Number? 1729 is the smallest natural number that can be written in the form of $a^3 + b^3$ in two different ways, a and b are also natural numbers <= 15. Guess the values of a and b, and print the two forms of $a^3 + b^3$. (Hint: use math.pow(a,3) to find cube of a)

4.
    a. Print the resulting evaluations of the two expressions 50-3*10 and (50-3)*10, on the same line.
    b. Print the logarithm of 4096 to the base 2.
    c. Print the roots of the quadratic equation: $2x^2 - 7x + 6$, on the same line, in ascending order (use the formula, roots = $\frac{-b \pm \sqrt{b^2 - 4ac)}}{2a}$). Verify your

answers.

5.

a. Convert -14.125 to binary number using the formulae on slide 53 here (Hint: sign bit is easy i.e. just use 0 for +ve no. and 1 for -ve no., divide the remaining part by powers of two till you get a decimal number of form 1 + pure decimal number (of form 0.xyzw….) and thus you can derive e-1023 as the power of two. Then, convert the pure decimal number 0.xyzw…. as a sum of powers of ½. To achieve this: subtract 0.5 and type the first bit (of 52 remaining bits) as 1 if the result is >=0 (else leftmost bit is 0), then subtract 0.25 and see if the result is >=0 and put the second bit as 1, else 0, and so on …)

- Here is solution for floating point number 3.5:
    - print(3.5/2^1) gives 1.75, hence e-1023 = 1, e = 1024, i.e., 10000000000
    - print(0.75 - 0.5) gives 0.25, result >= 0, so the 1st bit of remaining 52 bits is 1
    - print(0.25 - 0.25) gives 0, result >= 0, so the 2nd bit of rem. 51 bits is 1
    - Since we have reached 0, all the remaining 50 bits are 0
    - i.e. 0.75 = 0.5 + 0.25 = ½ + $(½)^2$, hence the first bit of the remaining 52 bit number is 1, and the 2nd bit is also 1, and remaining bits are 0.
    - So the final number is 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 …
    - We can print the number using: print("11" + "0" * 10 + "11" + "0"*50),
    - and verify its length by using function: len("11" + "0" * 10 + "11" + "0"*50)
      (it should print 64)

b. Convert the binary number created in Q) 5 a again to floating point and verify your answer.