CS-101 **IF/ELSE and Loops** **Practice Questions**

1. Write pseudo code that adds first 10 natural numbers.
2. Write pseudo code to find largest of three numbers. You cannot use while loop in this problem.
3. Write a pseudo code to print a square of ‘*’ using single while loop and if-else conditions.
4. Write pseudo code to calculate average of first 10 natural numbers. Now re-write the same code for calculating average of 10 numbers taken as input from user.
5. Write pseudo code that calculates sum of all natural numbers below 1000 that are multiples of 3 or 5 but not both.
6. Write a pseudo code that calculates the factorial of a number, where n! = n * (n-1)!, if n = 4 then your code should output 24 = 4*3*2*1;
7. Prompt for and enter 55 test scores (integer values in the range 0-100), and print how many were passing scores (50 or more) and how many failing scores (49 or less).
8. Write pseudo code that calculates x power y.
9. Write a pseudocode to convert a binary number to decimal.
10. Write a pseudocode to convert a decimal number to binary.
11. Write a pseudocode that convert a decimal number to IEEE 32 bit floating point number.
12. Write a pseudocode to convert octal number to hexadecimal.
13. Write pseudocode to calculate greatest common divisor (GCD) of a number.
14. Write a pseudo code that takes as input a number and tells whether it is prime or not. (a prime number is a natural number greater than one that has no positive divisors other than one and itself).
15. When squirrels get together for a party, they like to have nuts. A squirrel party is successful when the number of nuts is between 40 and 60, inclusive. Unless it is the weekend, in which case there is no upper bound on the number of nus. Write pseudocode of a program which inputs the number of nuts consumed at a party, and prints “True” if the party with the given values is successful, or “False” otherwise.
16. You are driving a little too fast, and a police officer stops you. Write pseudocode to compute the result, encoded as an integer value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.
17. A subject has an assignment and an exam. Both the assignment and exam are marked out of 100. The assignment contributes 40% towards the assessment of the subject and the exam 60%. A

program is required that allows the user to input assignment marks (out of 100) and an exam marks (out of 100) and display overall marks in the proportions specified.

18. Emporium Mall gives a discount of 25% on Independence Day to customers. The program is to display the discount and the amount due. [Think of inputs by yourself]

19. Write a loop to print this *fencepost* pattern. Such *fencepost* loops can be created by placing one *post* (i.e. |) outside your loop, and then alternating between *wires* (i.e. ==) and *posts* inside the loop.

```
|==|==|==|==|
```

20. Write pseudocode for a program that takes an integer and uses a fencepost loop to print the factors of that number, separated by the word "and". For example, for the number 24, it should print the following output.

```
1 and 2 and 3 and 4 and 6 and 8 and 12 and 24
```

21. Write a program to output the individual digits of a number from right to left with spaces added in between each digit. For example if the number is 9867 then your program should output: 7 6 8 9

22. Repeat the above so that the digits are printed from left to right. For example for 9867 the output should be 9 8 6 7. The maximum number you have to cater for is 10^5. The leading zeros should NOT be printed so if the number is 105 the output should be 1 0 5 and NOT 0 0 0 1 0 5.

23. Write a program that inputs numbers from the user till the user inputs a negative number. The program should then print the total number of times the number 1 was input.

24. Write a program that inputs numbers from the user till the user inputs a negative number. The program should then print the maximum of all numbers and minimum of all numbers that were input. The last negative number should not count towards the minimum. For example if the input is: 10 7 42 2 2 0 56 6 -1

The output should be: maximum is 56 and minimum is 0

25. Write a program that inputs numbers from the user till the user inputs a negative number. The program should then print the second maximum of all numbers and second minimum of all numbers that were input. The last negative number should not count towards the minimum. For example if the input is: 10 7 42 2 2 0 56 6 -1

The output should be: second maximum is 10 and second minimum is 2