# Worksheet 2

**General Instructions**: Do not copy-paste from this file to terminal. If you have doubts, contact the instructors or TAs. And **do not panic**!

- Each problem in this worksheet will require you to write a program.
- You should keep all your files in `CS1101/ws02` folder.
- Use `gedit` or `nano` to type your programs.
- The name of the programs should be `prob-n.py` for $\mathrm{n}^{\mathrm{th}}$ problem.
- Save the output of your program in a text file `prob-n-output.txt`.
- After you finish, create an archive of the folder `ws02` with name `ws02-idnumber.tgz` and upload in `WeLearn`.

1. a) Store a string: `The quick brown fox jumps over the lazy dog` in a variable `x`.
   b) Check whether the word `fox` is in this sentence.
   c) Print the characters in reverse order.
   d) Print every third character of the above sentence.

2. `range(start, end, step)` is built-in function in Python that returns a sequence of numbers from `start` to `stop` in steps of `step`. For example, the following code prints all the integers from 1 to 9 in different lines.

```
for i in range(1,10):
    print(i)
```

   a) Modify the above program such that for any value of $n$, it will generate a list of square of integers $1$ to $n$. Choose any value for the integer variable $n$ and the output should look like:

```
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
11 121
:
:
```

   BONUS: Create two variables that will store the sum of the numbers and the sum of the square of the numbers respectively. Print them at the end of the program.

b) i) Store 3.1415 in a variable $x$

ii) Store 22 in a variable $y$

iii) Store 7 in a variable $z$

iv) Check the output of $x/y$, $y/z$ and $z/x$. Are all of them float?

v) The following code snippet outputs: $3 \quad 3.14$

```
print('%4d'%x,'%7.2f'%(y/z))
```

Change the above code to produce the following output:

$$3.142 \;\; 3.142857$$

vi) In the following print statement, 4 places have been reserved for variable `x` and 5 places for variable `y`. This is called formatted printing.

```
print('%4d'%x,'%5d'%y)
```

Initialise a variable $n$ to an integer of your choice. Use formatted printing to obtain an output such as the following (example output when $n = 12$):

```
   1     1
   2     4
   3     9
   4    16
   5    25
   6    36
   7    49
   8    64
   9    81
  10   100
  11   121
  12   144
```

3. a) Initialise two variables `fname`, `mname` and `lname` with your first, middle and last name respectively. Initialise another variable `roll` with your roll number.

b) Print intials of your name. For example, if your name is *Arthur Conan Doyle*, your program should print: *ACD*. If your name is *Sherlock Holmes*, your program should print: *SH*.

c) Write another program to generate your IISERK ID number. If your name is *Arthur Conan Doyle* and your roll number is: $93$, your program should generate the ID as: $acd23ms093$.

4. Using `for` loops, write a program to print the following pattern :

```
abcdefghijkl
abcdefghijk
abcdefghij
abcdefghi
abcdefgh
abcdefg
abcdef
abcde
abcd
abc
ab
a
```

5. The following program prints the first 10 numbers of an AP series:

```
a = 2 # initial value
b = 5 # increment
for i in range(10):
    print(’%3d %6d’ %(i+1, a))
    a = a + b
```

In the above code, the part after # mark, is called a comment and is ignored by the shell. Adding comment helps increase the usefulness of a program (and helps *fixing* or *debugging* it).

Modify this program to print a geometric progression $a, ar, ar^2, ar^3, \ldots$ with an inital value $a = 2$ and the common ratio $r = 3$.

6. A Fibonacci sequence is given by $1, 1, 2, 3, 5, 8, 13, \ldots$. The sequence can be generated by $a_n = a_{n-1} + a_{n-2}$, where, $a_n$ is the $n^{\text{th}}$ entry in the sequence. It is given then $a_1 = a_2 = 1$. Using a `for` loop generate a Fibonacci sequence having a total of 30 values. Also print a formatted output having the serial number in the sequence and the corresponding Fibonacci number, as shown below:

```
1  1
2  1
3  2
4  3
5  5
6  8
7 13
8 21
```

7. Consider the following program. It uses nested `for` loops.

```
for i in range(2):
    for j in range(3):
        print(i, j, i+j)
print(’Done’)
```

Run the above program and observe the output. Using the concept of the nested `for` loops (taking help from the above code), write a program to calculate the double summation and print the result.

$$\sum_{n=1}^{n=5}\sum_{m=1}^{m=5}\frac{(n-m)^2}{(n+m)^2}.$$

Modify the above code to calculate the following double summation and print the result.

$$\sum_{n=1}^{n=5}\sum_{m=n+1}^{m=5}\frac{(n-m)^2}{(n+m)^2}.$$

BONUS: Compare the number of times the loops are executing.