Experiment 1

Problem Statement:

Q.1 Write a python program to print following lines in specific format Using only one print() function

```
Twinkle Twinkle Little Star,
        "How I wonder what you are!"
                Up above the world so high
                Like a diamond in the sky.
'Twinkle Twinkle' Little star
        How I wonder what you are
```

Theory

In this question we will learn about long string literals and escape sequences in python.
Long string literals are just like normal strings, but they span over multiple lines and print exactly what their content is. They are written inside 3 quotes or 3 double quotes.

Example:

```
x = “””

This is a long string literal.
    I hope this makes it clear for you!
        Have a nice day.

“””
```

Escape Characters:

To insert characters that are illegal in a string, use an escape character.
An escape character is a backslash \ followed by the character you want to insert.
An example of an illegal character is a double quote inside a string that is surrounded by double quotes

Example:

```
intro = “My name is “Dheeraj”. I am “19” years old.”
```

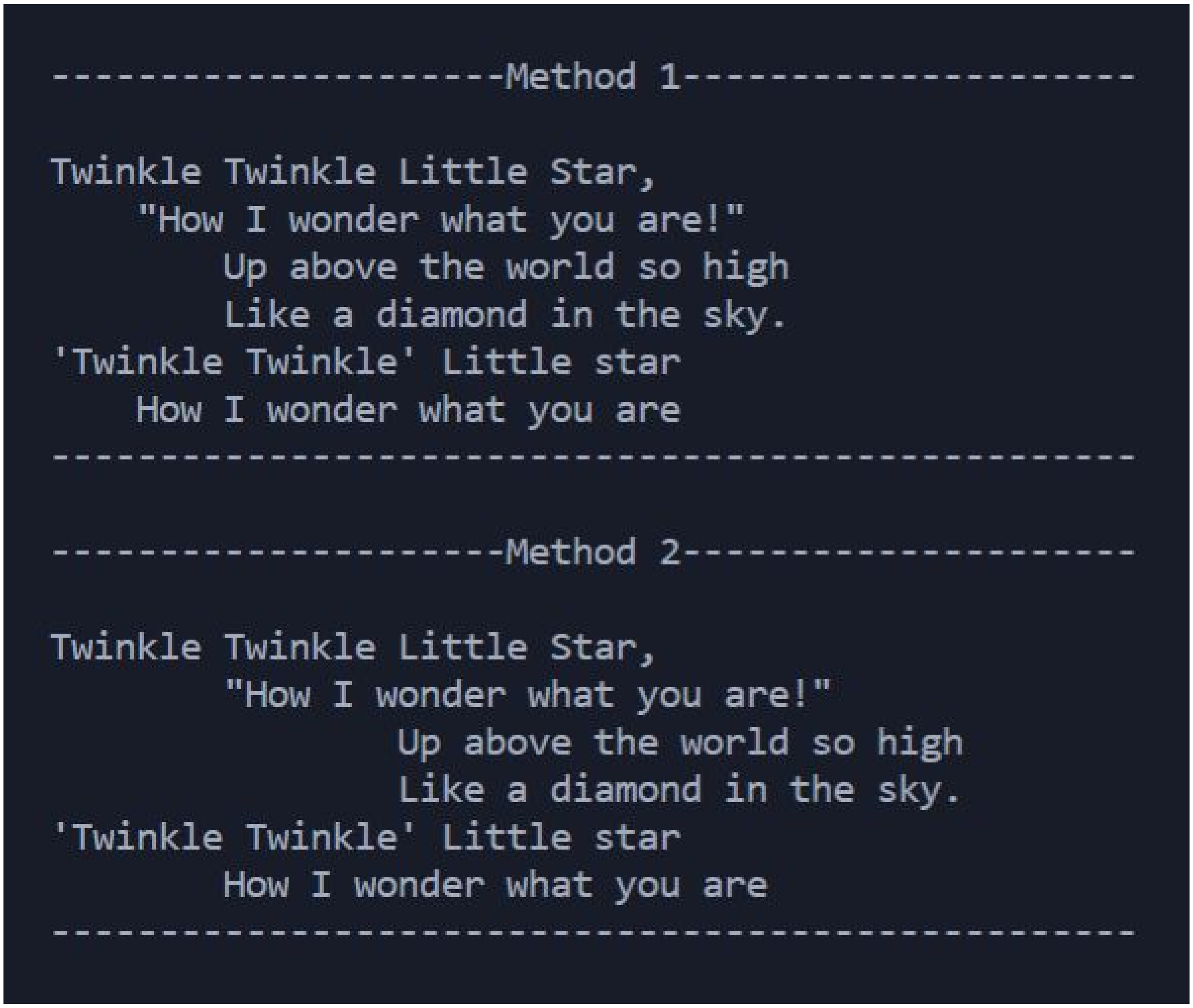
The correct way to write this will be:

```
intro = “My name is \“Dheeraj\”. I am \“19\” years old.”
```

Code

```
print("\n\n")

print("------------------Method 1------------------\n")

following_lines = '''
Twinkle Twinkle Little Star,
    "How I wonder what you are!"
        Up above the world so high
        Like a diamond in the sky.
'Twinkle Twinkle' Little star
    How I wonder what you are '''

print(following_lines)

print("---------------------------------------------\n")

print("------------------Method 2------------------\n")

print("Twinkle Twinkle Little Star,\n\t\"How I wonder what
you are!\"\n\t\tUp above the world so high\n\t\tLike a
diamond in the sky.\n\'Twinkle Twinkle\' Little star\n\tHow
I wonder what you are")

print("---------------------------------------------\n\n")
```

Output



```
---------------------Method 1---------------------

Twinkle Twinkle Little Star,
    "How I wonder what you are!"
        Up above the world so high
        Like a diamond in the sky.
'Twinkle Twinkle' Little star
    How I wonder what you are
--------------------------------------------------

---------------------Method 2---------------------

Twinkle Twinkle Little Star,
        "How I wonder what you are!"
                Up above the world so high
                Like a diamond in the sky.
'Twinkle Twinkle' Little star
        How I wonder what you are
--------------------------------------------------
```

Problem Statement:

Q.2 Program to show output formatting take two values and display them using single print() function

•Str.format()
•% operator

Theory

In this question we will learn about string formatting in python.
The format function helps us insert variable values in a string dynamically.
Python also uses C-style string formatting to create new, formatted strings. The "%" operator is used to format a set of variables enclosed in a "tuple" (a fixed size list), together with a format string, which contains normal text together with "argument specifiers", special symbols like "%s" and "%d".

Syntax:

1] str.format( )

```
name = “Dheeraj Lalwani”
age = 19
intro = “My name is {}. My age is {}. ”.format(name, age)
```

We can also do it this way:

```
intro = f“My name is {}. My age is {}.”
```

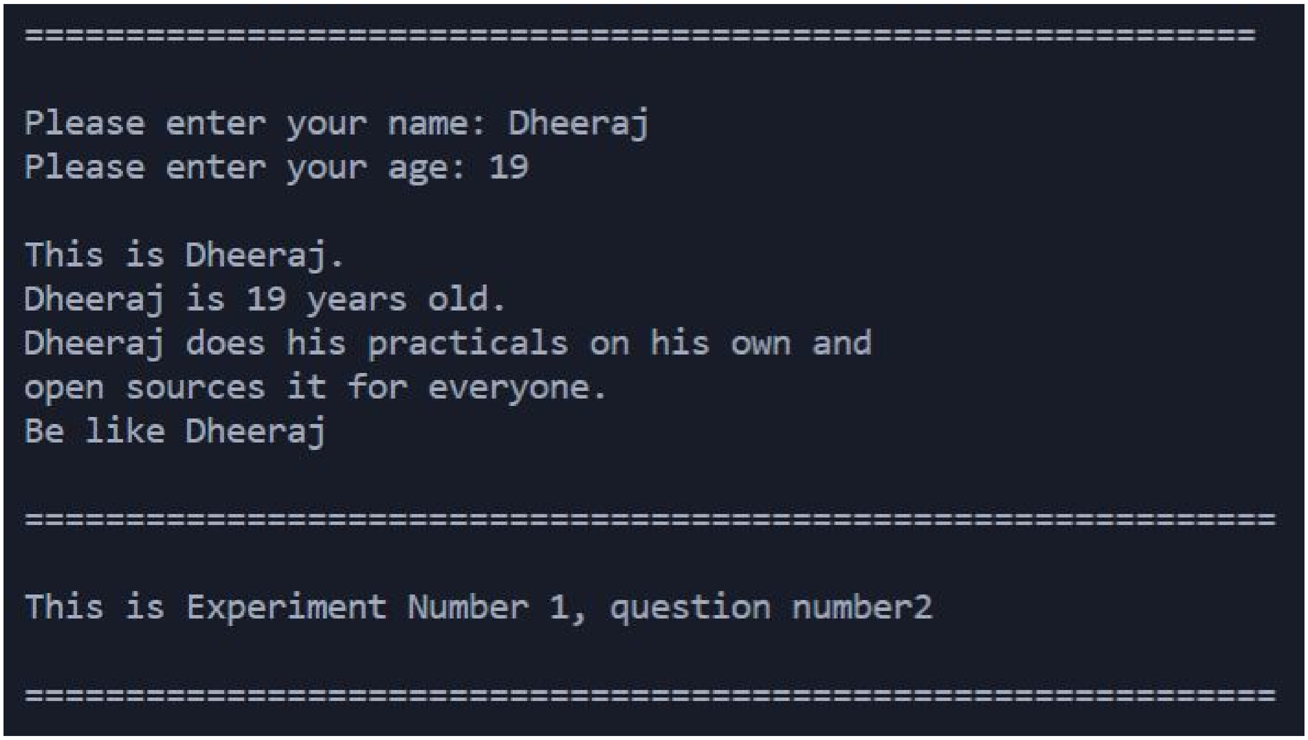
2] %d, %f, %s and other format specifiers

```
name = “Dheeraj Lalwani”
age = 19
intro = “My name is %s. My age is %d.” % (name, age)
```

Code

```
print("\n========================================\n")

name = input("Please enter your name: ")

age = int(input("Please enter your age: "))

print("\nThis is {}.\n{} is {} years old.\n{} does his
practicals on his own and \nopen sources it for
everyone.\nBe like {}".format(
    name, name, age, name, name))

print("\n========================================\n")

questionNumber = 2

experimentNumber = 1

print("This is Experiment Number %d, question number%d" %
      (experimentNumber, questionNumber))

print("\n========================================\n")
```

Output



```
=============================================================


Please enter your name: Dheeraj
Please enter your age: 19

This is Dheeraj.
Dheeraj is 19 years old.
Dheeraj does his practicals on his own and
open sources it for everyone.
Be like Dheeraj

==============================================================


This is Experiment Number 1, question number2

==============================================================
```

Problem Statement:

Q.3 Program to find whether input year is leap year or not using 'nested if

Theory

- In this question we will learn about if….else ladders.
- During the execution of the program we may not wish to execute all sets of statements sequentially.
- It is a conditional statement used for selection between multiple sets of statements based on multiple test conditions.
- The various test conditions are provided inside each if statement.
- Whenever the test condition is evaluated as True, the statements inside the corresponding if block are executed and the control comes out of the else-if ladder. If none of the test conditions are evaluated as True, the statements inside the else block are executed.
- As we have multiple sets of statements to select based on the test conditions, it is also called a multi way selection statement
- In the else-if ladder the conditions are evaluated from the top of the ladder downwards. As soon as a true condition is found, the statement associated with it is executed skipping the rest of the ladder.

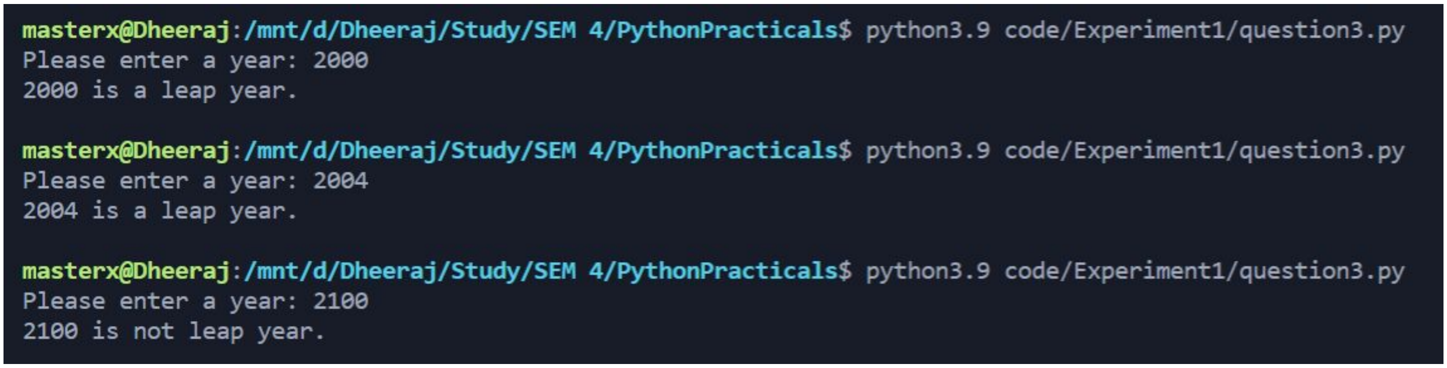
Syntax:

```
if(name == “Dheeraj Lalwani”):
    print(“Welcome Master.”)

elif(name == “Gayatri Joshi”):
    print(“The Master is expecting you, Gayatri Ji!”)

else:
    print(“Please get an appointment from the Master”)
```

Code

```
year = int(input("Please enter a year: "))

if year % 400 == 0:
    print(f"{year} is a leap year.\n")
elif year % 100 == 0:
    print(f"{year} is not leap year.\n")
elif year % 4 == 0:
    print(f"{year} is a leap year.\n")
else:
    print(f"{year} is not leap year.\n")
```

Output



```
masterx@Dheeraj:/mnt/d/Dheeraj/Study/SEM 4/PythonPracticals$ python3.9 code/Experiment1/question3.py
Please enter a year: 2000
2000 is a leap year.

masterx@Dheeraj:/mnt/d/Dheeraj/Study/SEM 4/PythonPracticals$ python3.9 code/Experiment1/question3.py
Please enter a year: 2004
2004 is a leap year.

masterx@Dheeraj:/mnt/d/Dheeraj/Study/SEM 4/PythonPracticals$ python3.9 code/Experiment1/question3.py
Please enter a year: 2100
2100 is not leap year.
```

Problem Statement:

Q.4 Program to print all armstrong numbers in the range 1 to 1000

Theory

Armstrong Number

- In this question we will learn about armstrong numbers and int to str type casting.
- Armstrong numbers are numbers such that the sum of its digits raised to the third power is equal to the number itself.

Example

371 is an Armstrong number, since `3**3 + 7**3 + 1**3 = 371`.

int to str type casting

We can type cast from int to str by using the `str()` function

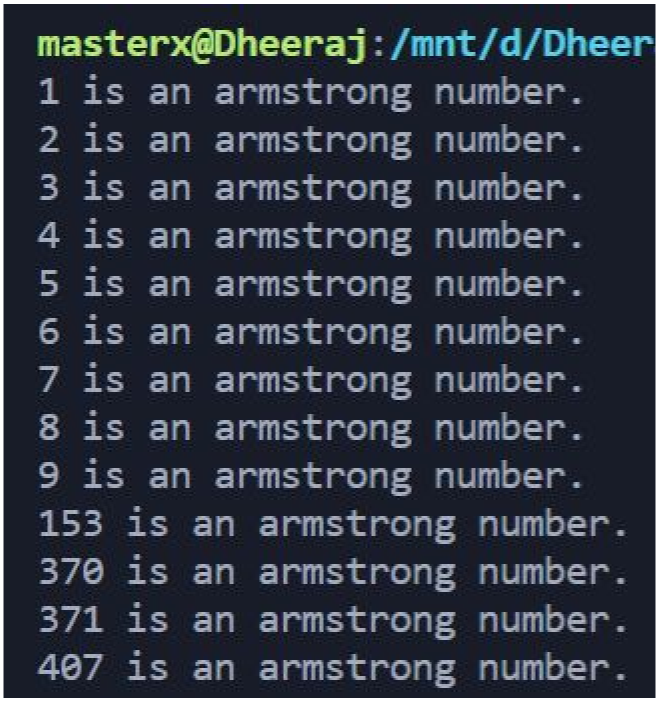
Example

```
x = 93843243
string_x = str(x)
```

Code

```
def isArmstrong(num):
    num = str(num)
    l = len(num)
    x = 0
    for digit in num:
        x += int(digit) ** l
    if(x == int(num)):
        return True

    return False


for i in range(1, 1001):
    if isArmstrong(i):
        print(f"{i} is an armstrong number.")
```

Output



```
masterx@Dheeraj:/mnt/d/Dheer
1 is an armstrong number.
2 is an armstrong number.
3 is an armstrong number.
4 is an armstrong number.
5 is an armstrong number.
6 is an armstrong number.
7 is an armstrong number.
8 is an armstrong number.
9 is an armstrong number.
153 is an armstrong number.
370 is an armstrong number.
371 is an armstrong number.
407 is an armstrong number.
```

Problem Statement:

Q.5 Program to find fibonacci series of n terms

Theory

Fibonacci series

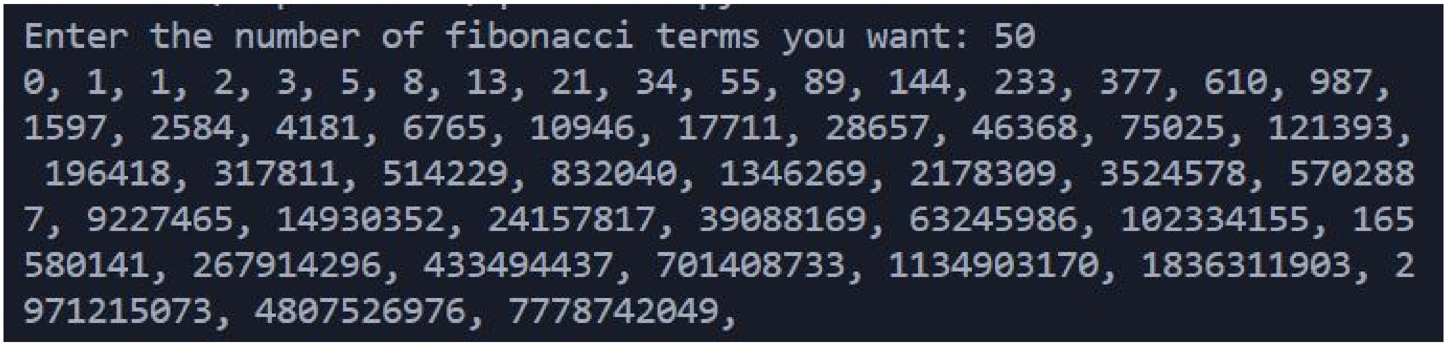
The Fibonacci sequence is a set of numbers that starts with a one or a zero, followed by a one, and proceeds based on the rule that each number (called a Fibonacci number) is equal to the sum of the preceding two numbers. If the Fibonacci sequence is denoted F (n), where n is the first term in the sequence, the following equation obtains for n = 0, where the first two terms are defined as 0 and 1 by convention:

F (0) = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 …

Code

```
n = int(input("Enter the number of fibonacci terms you
want: "))

a, b = 0, 1
for i in range(n):
    print(a, end=", ")
    a, b = b, a + b
```

Output



```
Enter the number of fibonacci terms you want: 50
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393,
 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 570288
7, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165
580141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2
971215073, 4807526976, 7778742049,
```

Problem Statement:

Q.6 Program to print pattern

Theory

Pattern Printing

Pattern printing helps us better understand the application of Loops and Nested Loops.
Loops are used when we want to run a particular piece of code over and over again.

For Loop In Python

The for loop works well with iterable objects like lists, tuples, strings, etc. This way, we can step through these object's items and manipulate their values based on our linking. The for loop is zero-indexed.

Syntax

```
x = [4, 3, 7, 1, 8, 5]
for i in x:
    print(i, “\n”)
```

While Loop In Python

The Python while loop executes a block of statements repeatedly as long as the condition is TRUE. We notice that it is a bit similar to the if statement. However, unlike the while loop, the if statement executes only once if its condition is TRUE.

Syntax

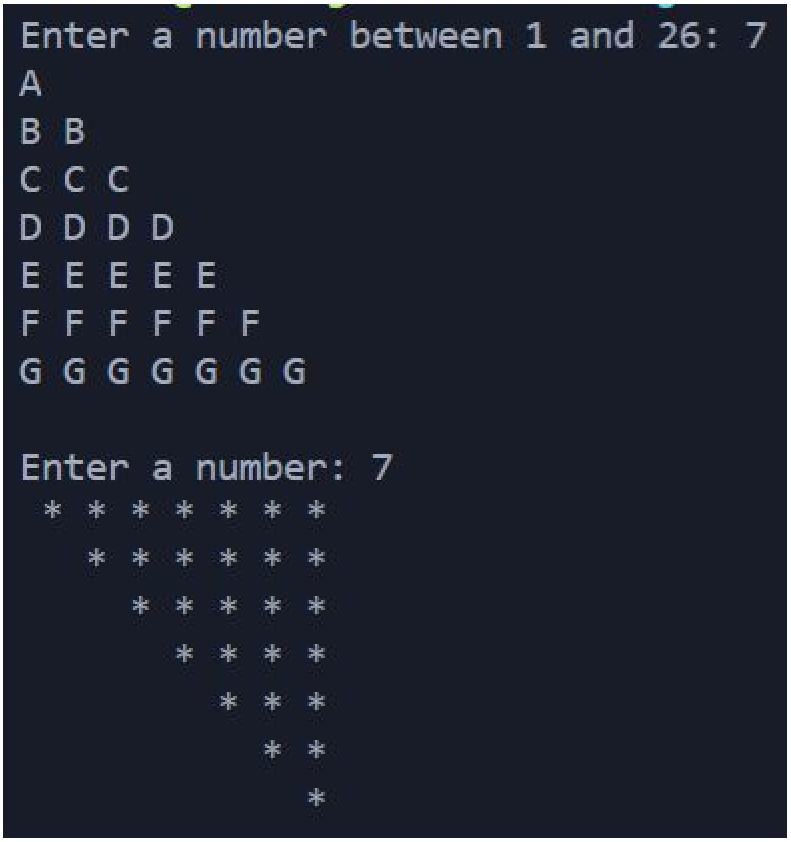
```
head = 10
water = 0
while head > water:
    print(“Water is not above head”)
    water += 1
```

Code

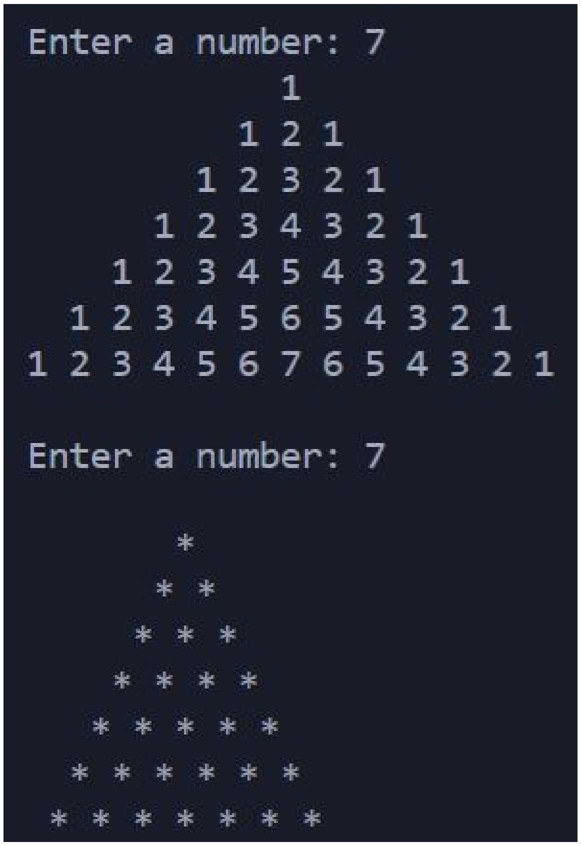
```
# 1]
# A
# B B
# C C C
# D D D D
# E E E E E


n = int(input("Enter a number between 1 and 26: "))
while(1 > n or n > 26):
    n = int(input("Please enter a number between 1 and 26: 
"))

char = 'A'
for num in range(1, n + 1):
    print(f"{char} " * num)
    char = chr(ord(char) + 1)


# 2]

# * * * * *
#   * * * *
#     * * *
#       * *
#         *


n = int(input("\nEnter a number: "))
for num in range(n):
    print("  " * (num), "* " * (n - num))


# 3]

#              1
#            1 2 1
#          1 2 3 2 1
#        1 2 3 4 3 2 1
#      1 2 3 4 5 4 3 2 1


n = int(input("\nEnter a number: "))
```

```
for num in range(1, n + 1):
    print("  " * (n - num), end="")
    for i in range(1, num + 1):
        print(i, end=" ")
    for i in range(num - 1, 0, -1):
        print(i, end=" ")
    print()


# 4]

#         *
#        * *
#       * * *
#      * * * *
#     * * * * *

n = int(input("\nEnter a number: "))

for num in range(n + 1):
    print(" " * (n - num), "* " * (num))
```

Output



```
Enter a number between 1 and 26: 7
A
B B
C C C
D D D D
E E E E E
F F F F F F
G G G G G G G

Enter a number: 7
 * * * * * * *
   * * * * * *
     * * * * *
       * * * *
         * * *
           * *
             *
```



```
Enter a number: 7
            1
          1 2 1
        1 2 3 2 1
      1 2 3 4 3 2 1
    1 2 3 4 5 4 3 2 1
  1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1

Enter a number: 7

       *
      * *
     * * *
    * * * *
   * * * * *
  * * * * * *
 * * * * * * *
```