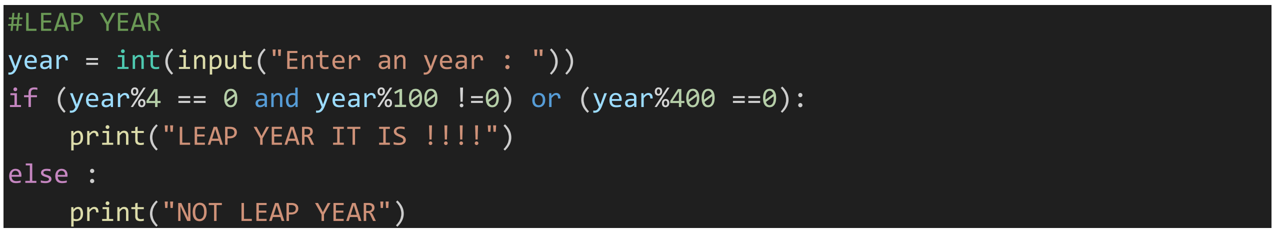
INTRODUCTION TO PYTHON

Lab sheet 1

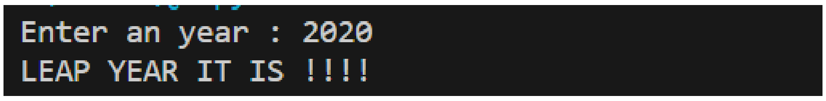
1. You are tasked with creating a program that determines whether a given year is a leap year or not. A leap year is a year that is exactly divisible by 4, except for years that are divisible by 100 but not by 400. Write a Python program that takes a year as input and prints whether it is a leap year or not.
   Conditions for Leap year
   - If a year is divisible by 4 and not divisible by 100, it is a leap year.
   - If a year is divisible by 100 and not divisible by 400, it is not a leap year.
   - If a year is divisible by 400, it is a leap year.

CODE:



```python
#LEAP YEAR
year = int(input("Enter an year : "))
if (year%4 == 0 and year%100 !=0) or (year%400 ==0):
    print("LEAP YEAR IT IS !!!!")
else :
    print("NOT LEAP YEAR")
```

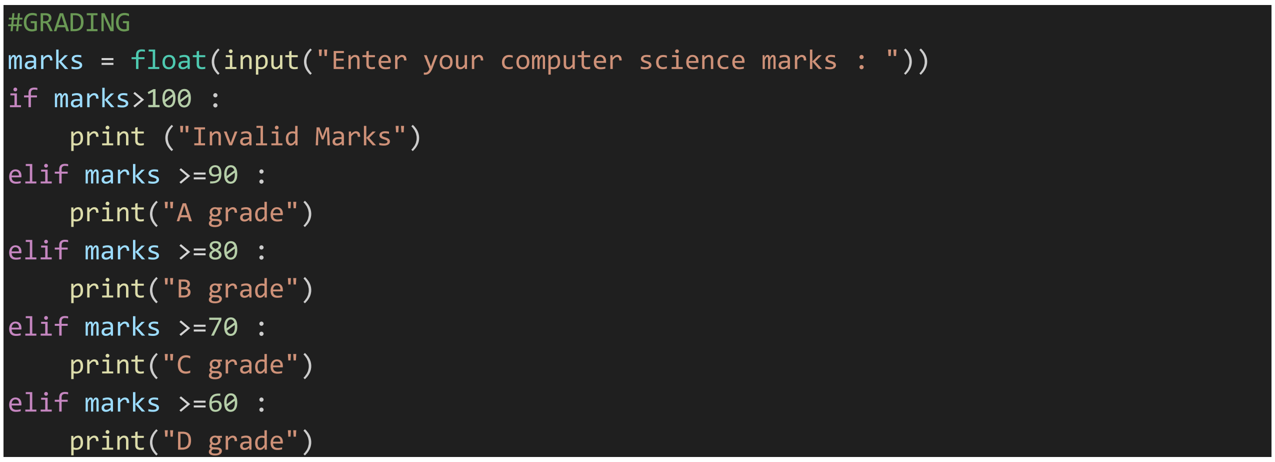
OUTPUT:



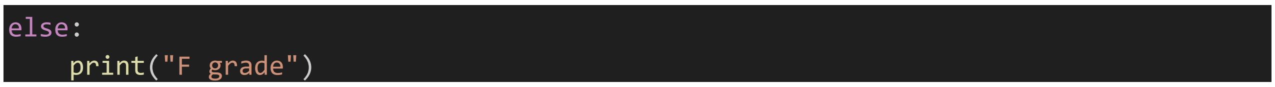
```
Enter an year : 2020
LEAP YEAR IT IS !!!!
```

2. You are responsible for grading the final exam of a computer science class. The grading scale is as follows:
   A: 90-100
   B: 80-89
   C: 70-79
   D: 60-69
   F: Below 60
   Write a Python program that takes a student's exam score as input and determines their grade using an if-else ladder. The program should display the grade earned by the student.

CODE:

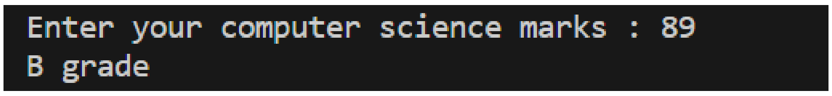


```python
#GRADING
marks = float(input("Enter your computer science marks : "))
if marks>100 :
    print ("Invalid Marks")
elif marks >=90 :
    print("A grade")
elif marks >=80 :
    print("B grade")
elif marks >=70 :
    print("C grade")
elif marks >=60 :
    print("D grade")
```



```
else:
    print("F grade")
```

OUTPUT:



```
Enter your computer science marks : 89
B grade
```

3. You are building a program to calculate the cost of shipping a package. The cost depends on the weight of the package and the distance it needs to be shipped. Here are the rules:

   • If the package weighs less than or equal to 2 pounds, the base cost is $5.00.

   • If the package weighs more than 2 pounds but less than or equal to 10 pounds, the base cost is $10.00.

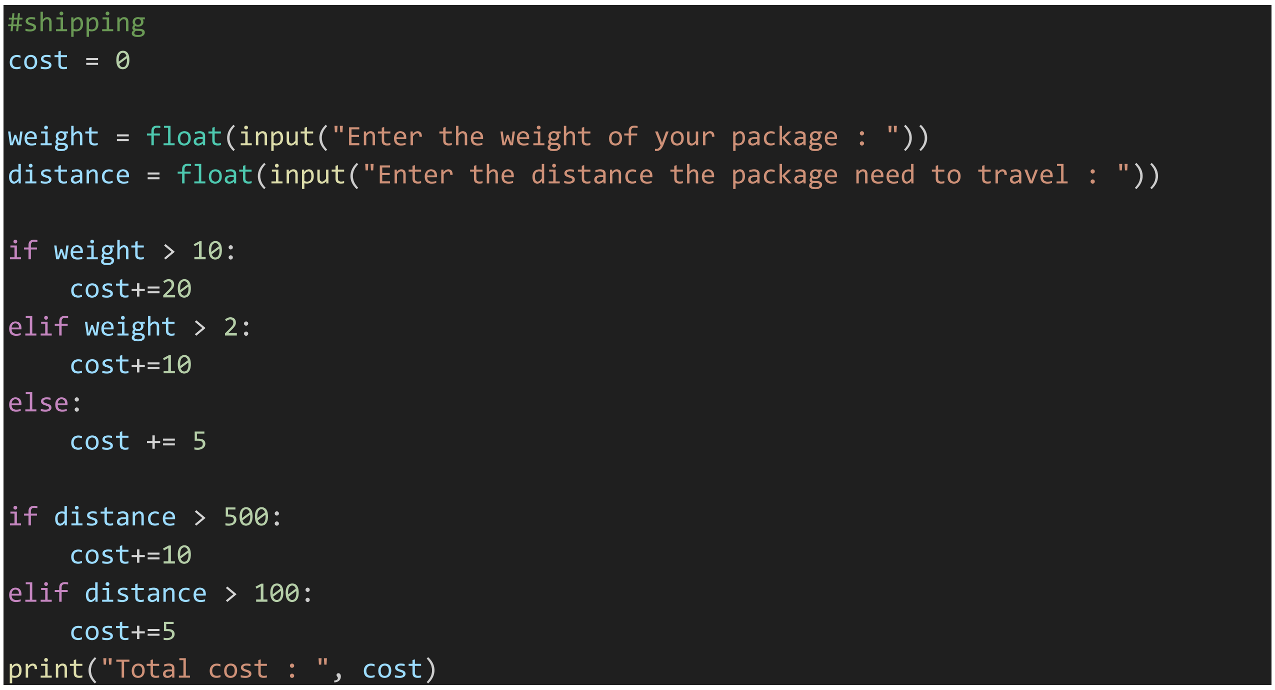
   • If the package weighs more than 10 pounds, the base cost is $20.00.

   • If the distance is less than or equal to 100 miles, there's no additional charge.

   • If the distance is greater than 100 miles but less than or equal to 500 miles, there's a $5.00 additional charge.

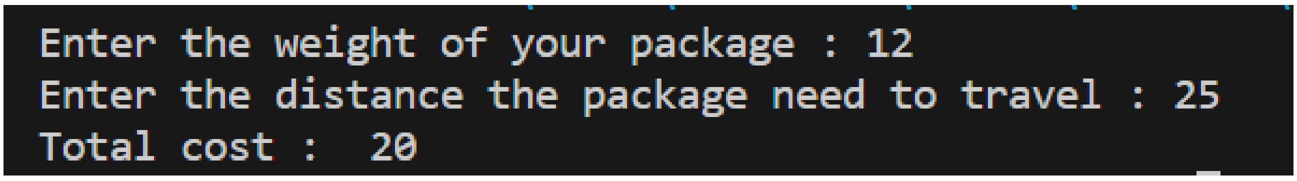
   • If the distance is greater than 500 miles, there's a $10.00 additional charge.

CODE:



```
#shipping
cost = 0

weight = float(input("Enter the weight of your package : "))
distance = float(input("Enter the distance the package need to travel : "))

if weight > 10:
    cost+=20
elif weight > 2:
    cost+=10
else:
    cost += 5

if distance > 500:
    cost+=10
elif distance > 100:
    cost+=5
print("Total cost : ", cost)
```

OUTPUT:

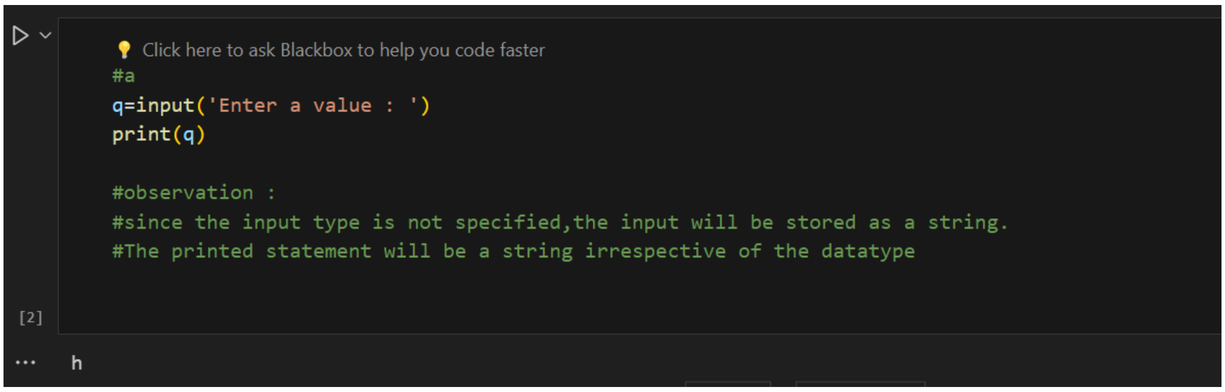


```
Enter the weight of your package : 12
Enter the distance the package need to travel : 25
Total cost :  20
```

4. Accepting user input. Write your observations of the output of (a) to (d)
   (a)

```
q=input('Enter a value: ')
print(q)
```

OBSERVATION AND OUTPUT:



💡 Click here to ask Blackbox to help you code faster

```
#a
q=input('Enter a value : ')
print(q)

#observation :
#since the input type is not specified,the input will be stored as a string.
#The printed statement will be a string irrespective of the datatype
```

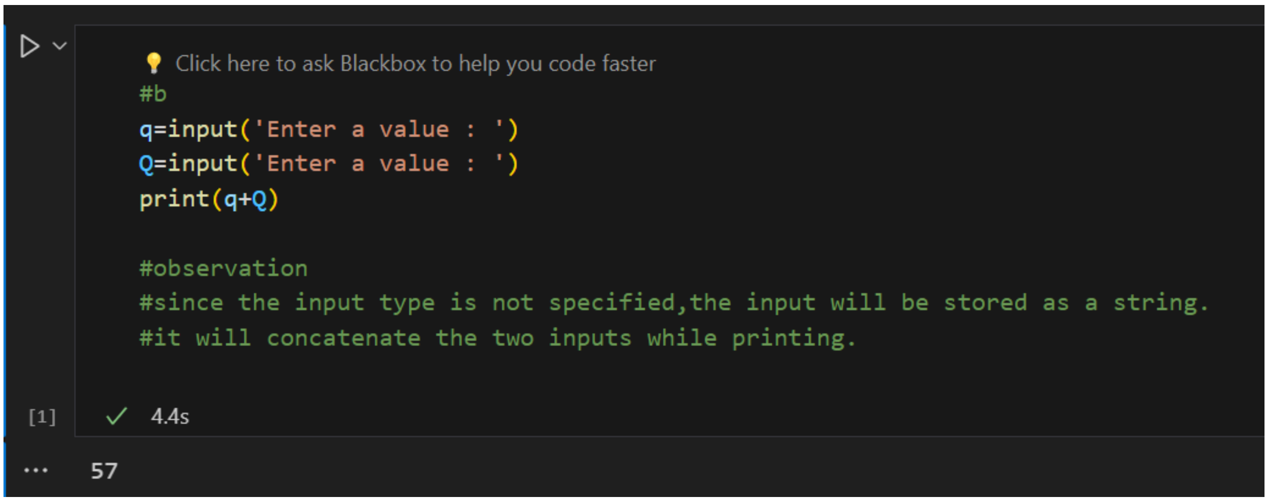
[2]

```
h
```

(b)

```
q=input('Enter a value: ')
Q=input('Enter a value: ')
print(q+Q)
```

OBSERVATION AND OUTPUT:



💡 Click here to ask Blackbox to help you code faster

```
#b
q=input('Enter a value : ')
Q=input('Enter a value : ')
print(q+Q)

#observation
#since the input type is not specified,the input will be stored as a string.
#it will concatenate the two inputs while printing.
```

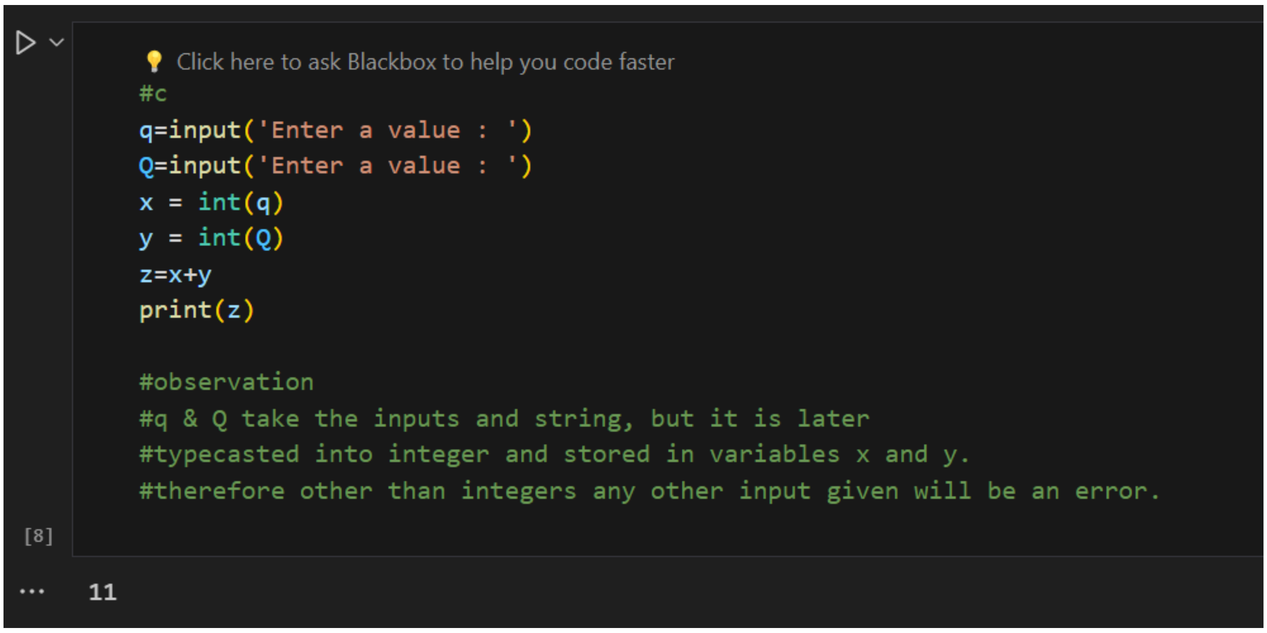
[1] ✓ 4.4s

```
57
```

(c)

```
q=input('Enter a value: ')
Q=input('Enter a value: ')
x=int(q)
y=int(Q)
z=x+y
print(z)
```

OBSERVATION AND OUTPUT:

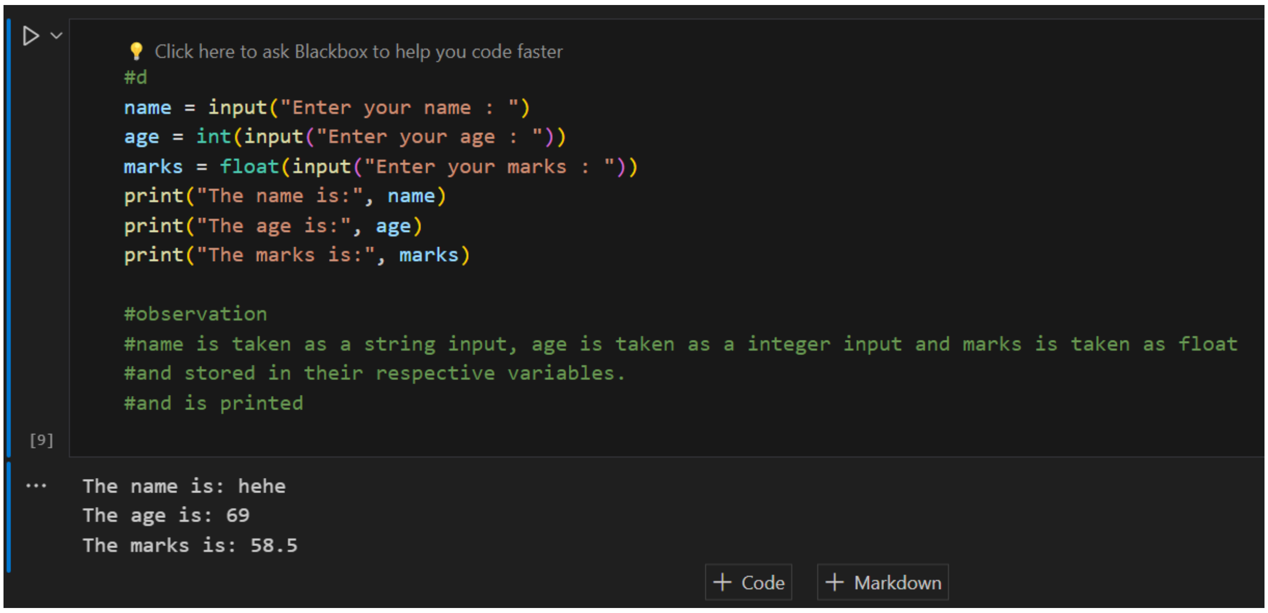


```
💡 Click here to ask Blackbox to help you code faster
#c
q=input('Enter a value : ')
Q=input('Enter a value : ')
x = int(q)
y = int(Q)
z=x+y
print(z)

#observation
#q & Q take the inputs and string, but it is later
#typecasted into integer and stored in variables x and y.
#therefore other than integers any other input given will be an error.
[8]
...  11
```

(d)

```
name = input("Enter your name: ")   # String Input
age = int(input("Enter your age: "))# Integer Input
marks = float(input("Enter your marks: ")) # Float Input
print("The name is:", name)
print("The age is:", age)
print("The marks is:", marks)
```

OBSERVATION AND OUTPUT:



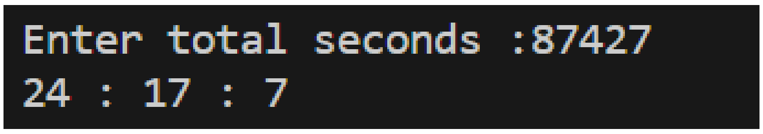
```
💡 Click here to ask Blackbox to help you code faster
#d
name = input("Enter your name : ")
age = int(input("Enter your age : "))
marks = float(input("Enter your marks : "))
print("The name is:", name)
print("The age is:", age)
print("The marks is:", marks)

#observation
#name is taken as a string input, age is taken as a integer input and marks is taken as float
#and stored in their respective variables.
#and is printed
[9]
...
The name is: hehe
The age is: 69
The marks is: 58.5
```

5. Write a program to read the number of seconds and print it in the form hr:min:sec.

CODE:

```
#Write a program to read the number of seconds and print it in the form
hr:min:sec.

seconds = int(input("Enter total seconds :"))

minute = seconds // 60
seconds -= minute*60
hour = minute // 60
minute -= hour*60

print(hour,":",minute,":",seconds)
```

OUTPUT:



```
Enter total seconds :87427
24 : 17 : 7
```

6. Which out of the code snippets below, print the numbers from 1 to 10. Give the reason for the error in the code snippets below which does not print from 1 to 10.

(a)

```
i=1
while i<10:
print(i)
i=i+1
```

(b)

```
i=1
while i<=10:
print(i)
i=i+1
```

(c)

```
i=3
while i<=10:
print(i)
i=i+2
```

(d)

```
i=1
while i<=10:
print(i)
i=i+1
```

(e)

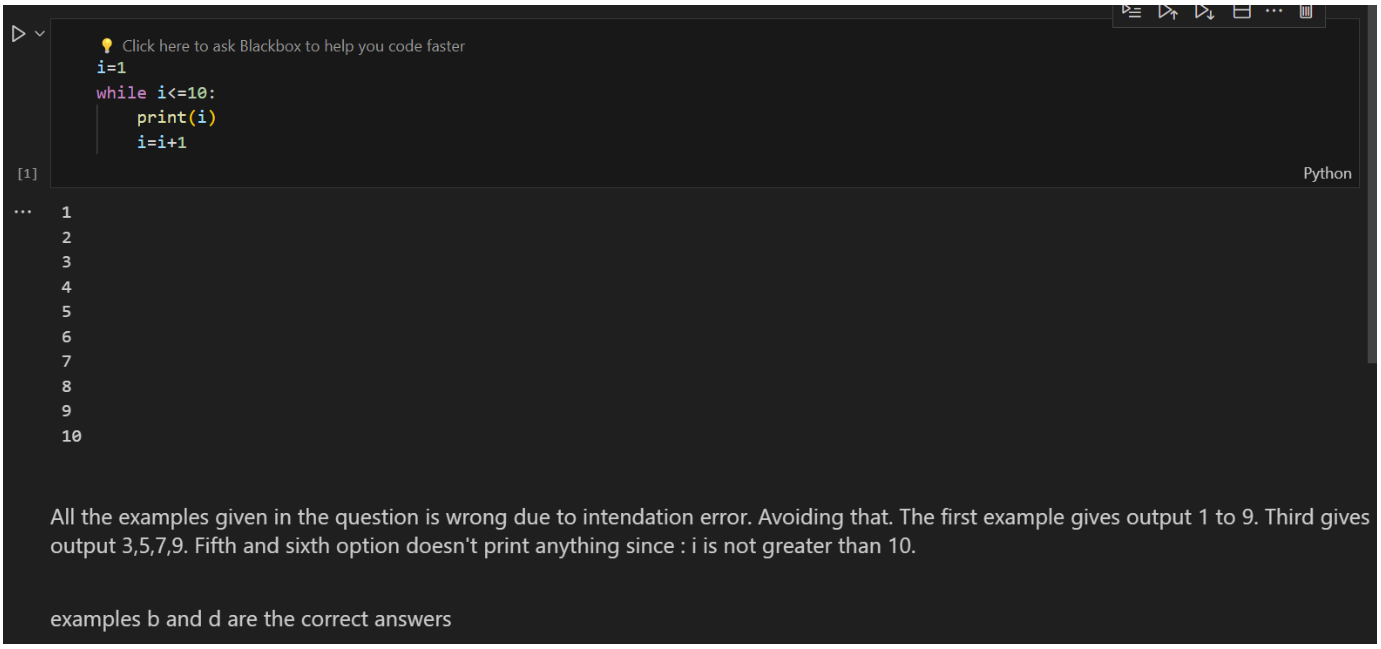
```
i=1
while i>=10:
```

```
print(i)
i=i+1
```

(f)

```
i=1
while i>=10:
print(i)
```

OBSERVATION :



Click here to ask Blackbox to help you code faster

```
i=1
while i<=10:
    print(i)
    i=i+1
```

[1] Python

```
1
2
3
4
5
6
7
8
9
10
```

All the examples given in the question is wrong due to intendation error. Avoiding that. The first example gives output 1 to 9. Third gives output 3,5,7,9. Fifth and sixth option doesn't print anything since : i is not greater than 10.

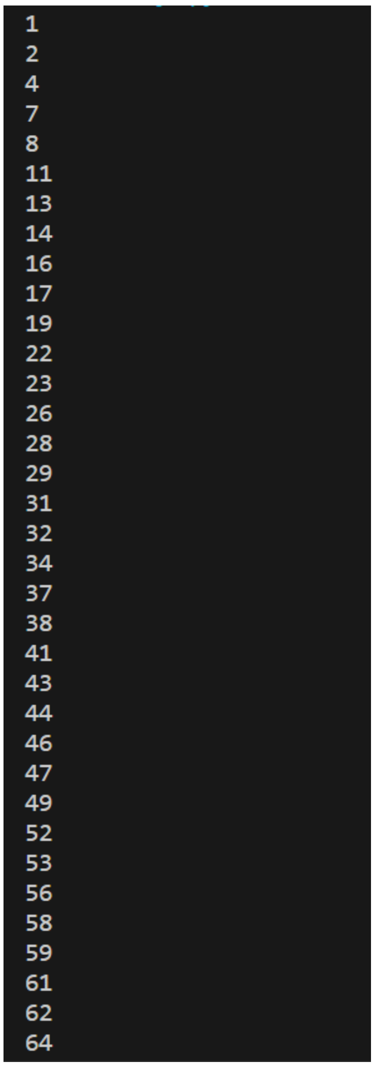
examples b and d are the correct answers

7. Write a Python program that prints all the numbers from 0 to 100 except multiples of 3 or 5.
   [Hint: Use continue statement.]

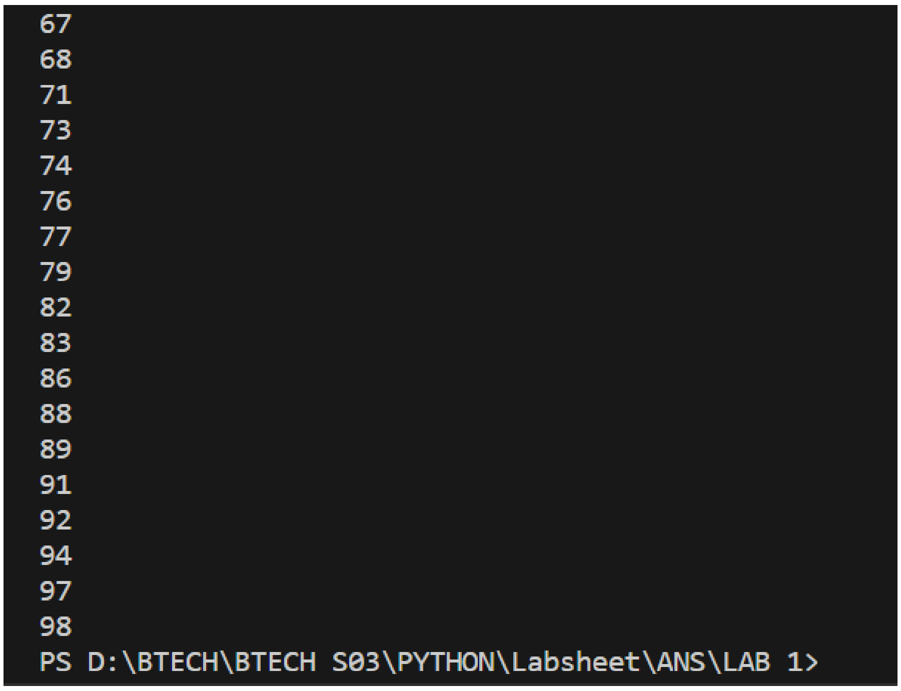
CODE:

```
#Python program that prints all the numbers from 0 to 100 except multiples of 3 or 5.

i = 0
while i<=100:
    if (i%3==0 or i%5==0):
        i+=1
        continue
    print(i)
    i+=1
```

OUTPUT:



```
1
2
4
7
8
11
13
14
16
17
19
22
23
26
28
29
31
32
34
37
38
41
43
44
46
47
49
52
53
56
58
59
61
62
64
```



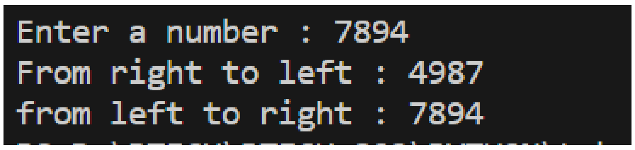
```
67
68
71
73
74
76
77
79
82
83
86
88
89
91
92
94
97
98
PS D:\BTECH\BTECH S03\PYTHON\Labsheet\ANS\LAB 1>
```

8. Write a Python program to take an n-digit integer and print the digits of the number from left to right and right to left.

CODE:

```python
#Python program to take an n-digit integer and print the digits of the number
from left to right and right to left.

number = input("Enter a number : ")

print("From right to left :", number[::-1])
print("from left to right :", number)
```

OUTPUT:



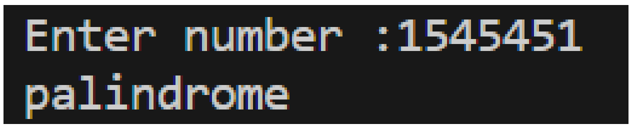
```
Enter a number : 7894
From right to left : 4987
from left to right : 7894
```

9. Write a python program to check if a number given by the user is a palindrome. (Hint: A number is a palindrome if the number is equal to its reverse.)

CODE:

```python
#python program to check if a number given by the user is a palindrome.

n = input("Enter number :")

if n[::-1] == n:
    print("palindrome")
else:
    print("not palindrome")
```

OUTPUT:



```
Enter number :1545451
palindrome
```

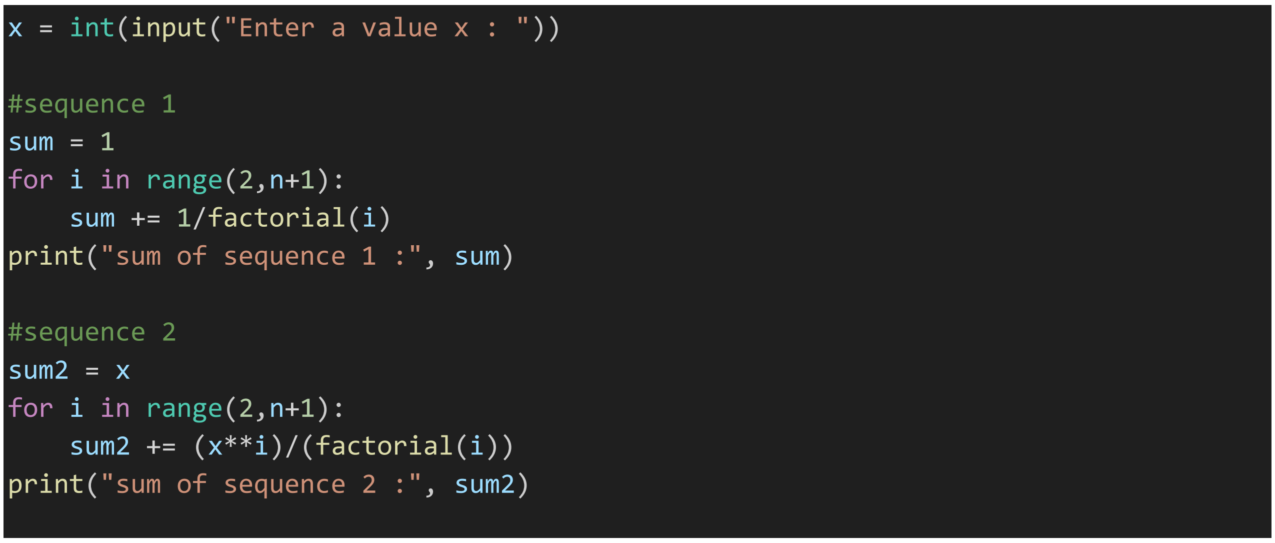
10. Write a Python program to find the sum of the below series provided n is a number given by the user.

$$1 + \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{1}{n!}$$

$$x + \frac{x^2}{2!} + \frac{x^3}{2!} + \ldots + \frac{x^n}{n!};$$

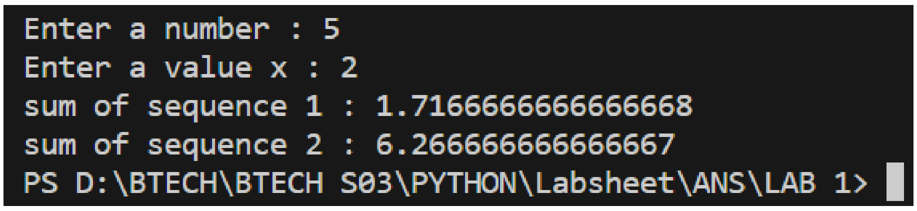
CODE:

```python
from math import factorial

n = int(input("Enter a number : "))
```



```python
x = int(input("Enter a value x : "))

#sequence 1
sum = 1
for i in range(2,n+1):
    sum += 1/factorial(i)
print("sum of sequence 1 :", sum)

#sequence 2
sum2 = x
for i in range(2,n+1):
    sum2 += (x**i)/(factorial(i))
print("sum of sequence 2 :", sum2)
```

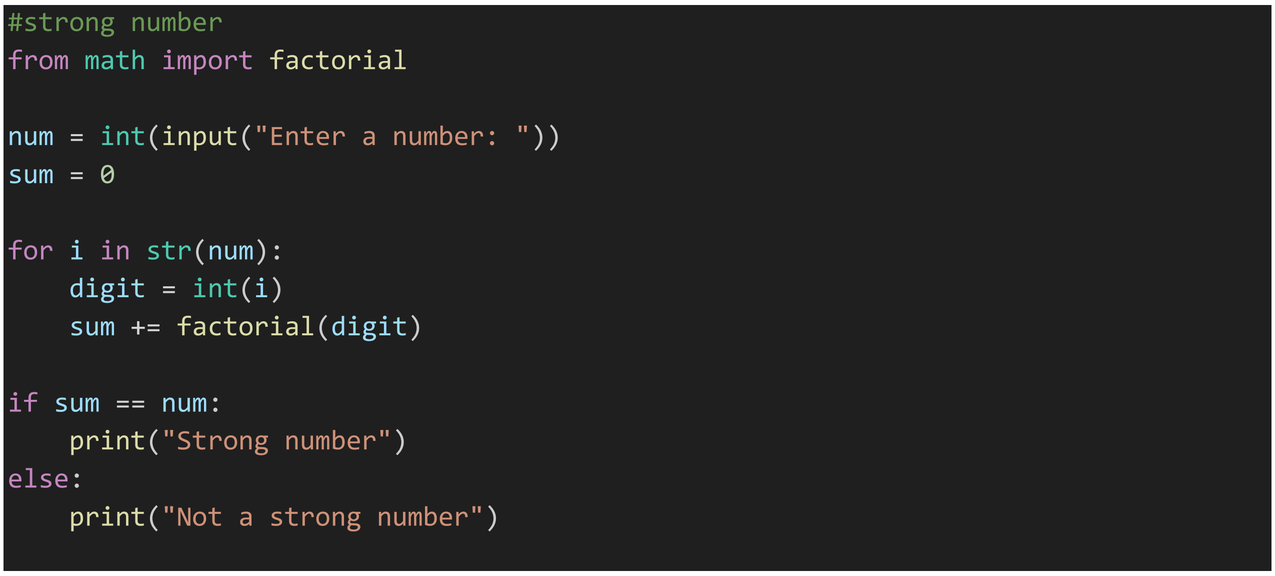
OUTPUT:



```
Enter a number : 5
Enter a value x : 2
sum of sequence 1 : 1.7166666666666668
sum of sequence 2 : 6.266666666666667
PS D:\BTECH\BTECH S03\PYTHON\Labsheet\ANS\LAB 1>
```

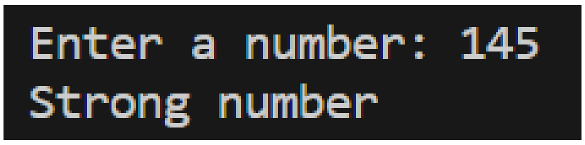
11. Write a program to check whether a number is strong number or not.*Strong number* is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since, 1! + 4! + 5! = 145

CODE:



```python
#strong number
from math import factorial

num = int(input("Enter a number: "))
sum = 0

for i in str(num):
    digit = int(i)
    sum += factorial(digit)

if sum == num:
    print("Strong number")
else:
    print("Not a strong number")
```

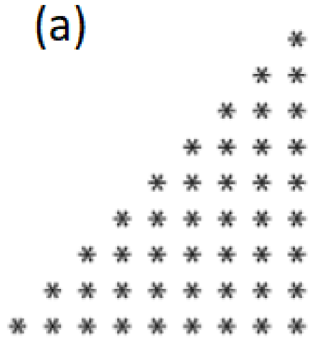
OUTPUT:



```
Enter a number: 145
Strong number
```

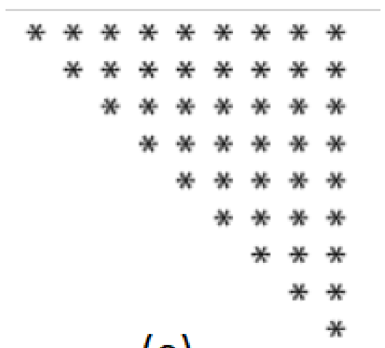
12. Write python program to print the below patterns. Take as input no: of rows

(a)



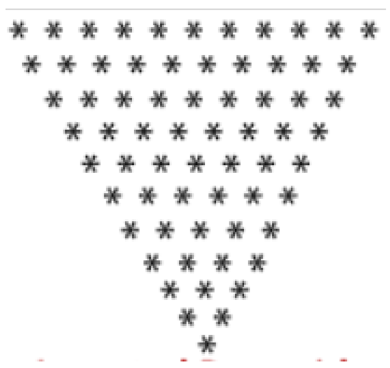
```
                *
              * *
            * * *
          * * * *
        * * * * *
      * * * * * *
    * * * * * * *
  * * * * * * * *
* * * * * * * * *
```

(b)



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

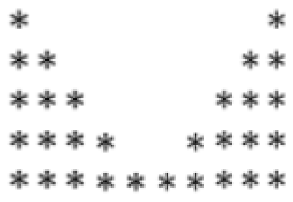
(c)



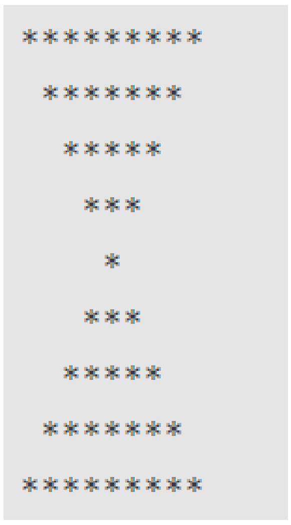
```
* * * * * * * * * * *
 * * * * * * * * * *
  * * * * * * * * *
   * * * * * * * *
    * * * * * * *
     * * * * * *
      * * * * *
       * * * *
        * * *
         * *
          *
```

(e)

(d)

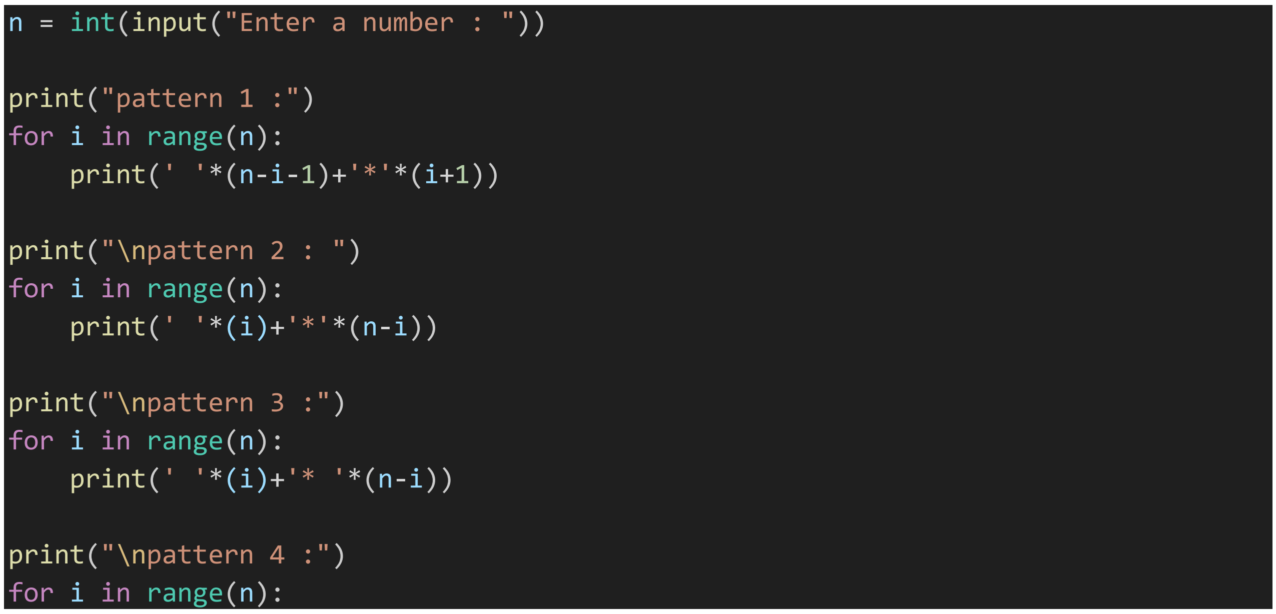


```
*                 *
* *             * *
* * *         * * *
* * * *     * * * *
* * * * * * * * * *
```

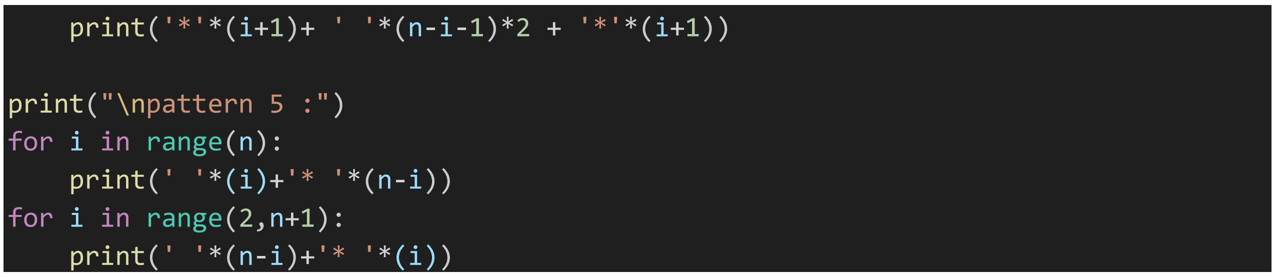


```
*********
 *******
  *****
   ***
    *
   ***
  *****
 *******
*********
```

CODE:

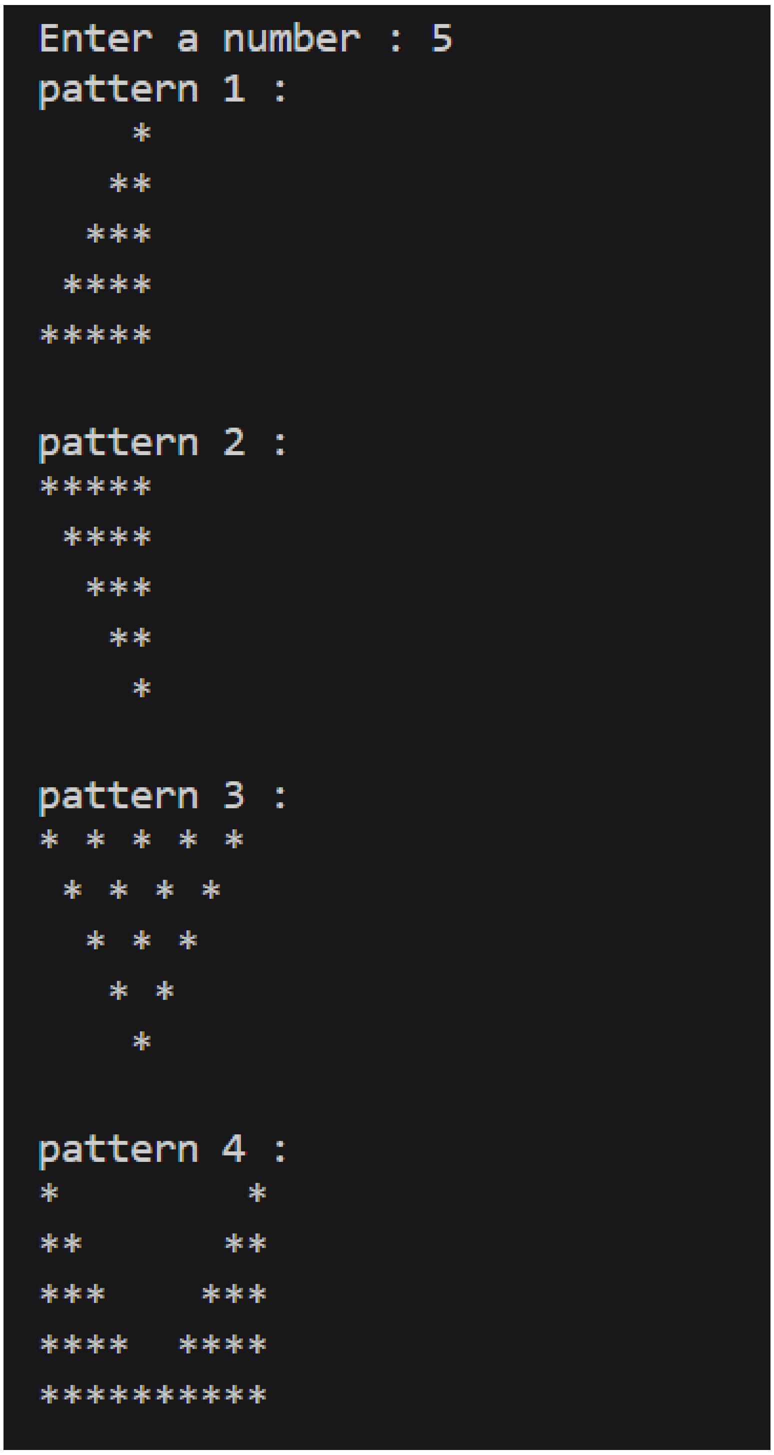


```
n = int(input("Enter a number : "))

print("pattern 1 :")
for i in range(n):
    print(' '*(n-i-1)+'*'*(i+1))

print("\npattern 2 : ")
for i in range(n):
    print(' '*(i)+'*'*(n-i))

print("\npattern 3 :")
for i in range(n):
    print(' '*(i)+'* '*(n-i))

print("\npattern 4 :")
for i in range(n):
```

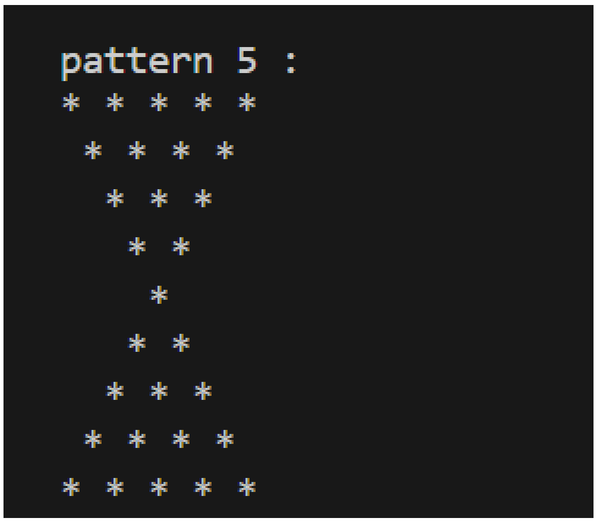


```python
    print('*'*(i+1)+ ' '*(n-i-1)*2 + '*'*(i+1))

print("\npattern 5 :")
for i in range(n):
    print(' '*(i)+'* '*(n-i))
for i in range(2,n+1):
    print(' '*(n-i)+'* '*(i))
```

OUTPUT:



```
Enter a number : 5
pattern 1 :
    *
   **
  ***
 ****
*****

pattern 2 :
*****
 ****
  ***
   **
    *

pattern 3 :
* * * * *
 * * * *
  * * *
   * *
    *

pattern 4 :
*        *
**      **
***    ***
****  ****
**********
```



```
pattern 5 :
* * * * *
 * * * *
  * * *
   * *
    *
   * *
  * * *
 * * * *
* * * * *
```

13. Write a Python program to print the below patterns.

(a)
```
1
12
123
1234
12345
```

(b)
```
    1
   12
  123
 1234
12345
```

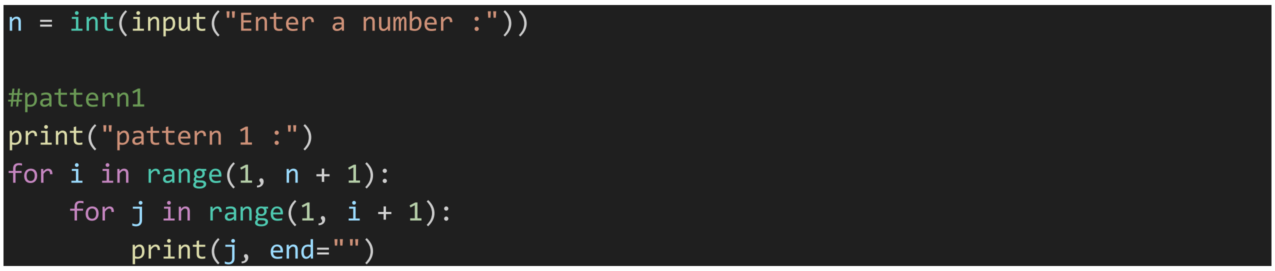
(c)
```
    11
   1221
  123321
 12344321
1234554321
```

(d)
```
12345
1234
123
12
1
```

(e)
```
    1
   121
  12321
 1234321
123454321
```

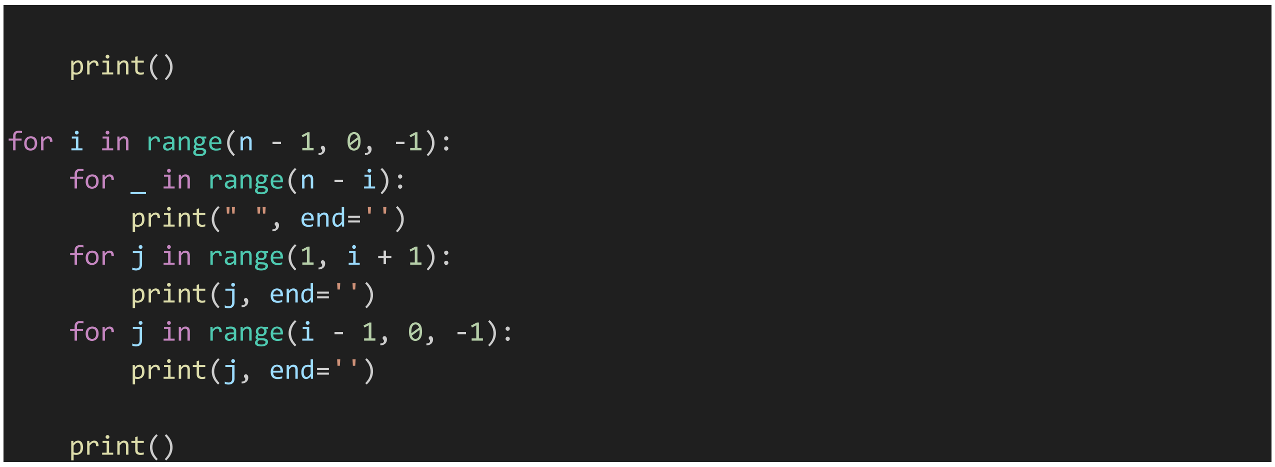
(f)
```
  1
 121
12321
 121
  1
```

CODE:



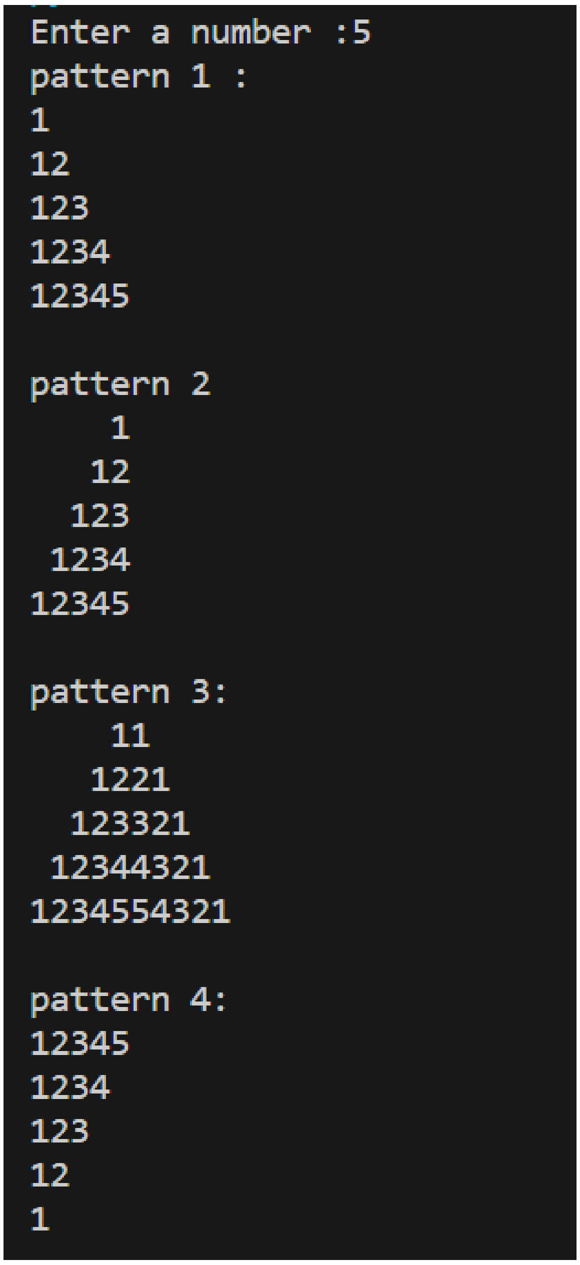
```
n = int(input("Enter a number :"))

#pattern1
print("pattern 1 :")
for i in range(1, n + 1):
    for j in range(1, i + 1):
        print(j, end="")
```

```python
    print()

#pattern 2:
print("\npattern 2")
for i in range(1, n + 1):
    for _ in range(n - i):
        print(" ", end='')

    for j in range(1, i + 1):
        print(j, end='')
    print()

#pattern 3:
print("\npattern 3:")
for i in range(1, n+1):
    print(" "*(n-i), end="")
    for j in range(1, i+1):
        print(j, end="")
    for j in range(i, 0, -1):
        print(j, end="")
    print("")
#pattern 4:
print("\npattern 4:")
for i in range(n, 0, -1):
    for j in range(1, i + 1):
        print(j, end='')
    print()

#pattern 5:
print("\npattern 5:")
for i in range(1, n + 1):
    for _ in range(n - i):
        print(" ", end='')
    for j in range(1, i + 1):
        print(j, end='')
    for j in range(i - 1, 0, -1):
        print(j, end='')

    print()

#pattern 6:
print("\npattern 6 :")
for i in range(1, n + 1):
    for _ in range(n - i):
        print(" ", end='')
    for j in range(1, i + 1):
        print(j, end='')
    for j in range(i - 1, 0, -1):
        print(j, end='')
```

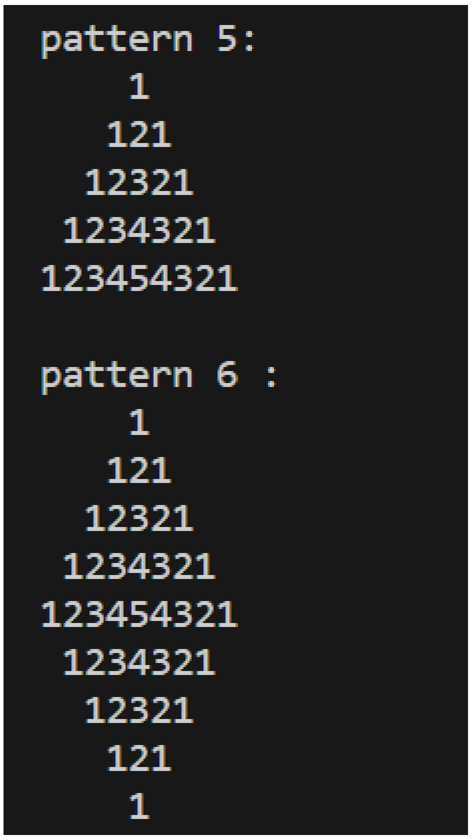


```
    print()

for i in range(n - 1, 0, -1):
    for _ in range(n - i):
        print(" ", end='')
    for j in range(1, i + 1):
        print(j, end='')
    for j in range(i - 1, 0, -1):
        print(j, end='')

    print()
```

OUTPUT:



```
Enter a number :5
pattern 1 :
1
12
123
1234
12345

pattern 2
    1
   12
  123
 1234
12345

pattern 3:
    11
   1221
  123321
 12344321
1234554321

pattern 4:
12345
1234
123
12
1
```



```
pattern 5:
    1
   121
  12321
 1234321
123454321

pattern 6 :
    1
   121
  12321
 1234321
123454321
 1234321
  12321
   121
    1
```