Tool & Technique Laboratory (T&T Lab.)

[CS-3096]

Individual Work

Lab. No: , Date: , Day:

Topic:

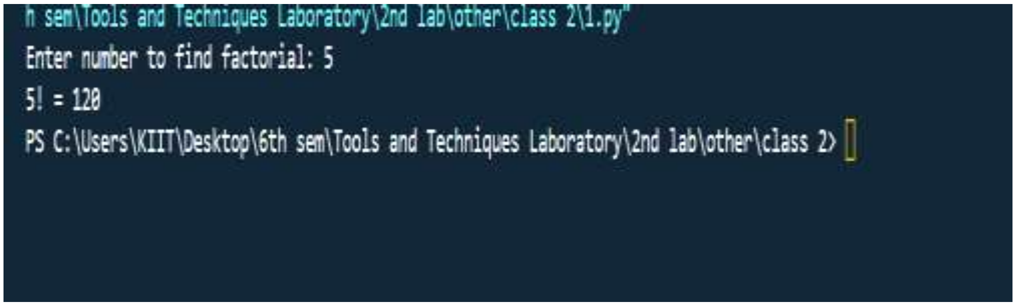
Roll Number:	20051939	**Branch/Section:**	**CSE - 17**
Name in Capital:	**SHASHIKANT SHAH**		

(Instruction: Rename this file as r-LAB-x where r is your roll number & x is your lab. number & Suppose your roll number is 1905123 & you want to submit lab-2 programs, then file name should be 1905123-LAB-2. Finally delete all texts inside parentheses, also parenthesis)

Program No: (1.)

Program Title:
(WAP to calculate the factorial of a given number)

Input/Output Screenshots:
RUN-1:
(Paste here the screenshots of first run)



```
h sem\Tools and Techniques Laboratory\2nd lab\other\class 2\1.py"
Enter number to find factorial: 5
5! = 120
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\class 2>
```

Source code
(Paste here the source code)

```
n = int(input("Enter number to find factorial: "))
prod = 1
for num in range(1, n + 1):
    prod *= num
print(f"{n}! = {prod}")
```

Conclusion/Observation
Successfully compile to calculate the factorial of a given number

Program No: (2.)

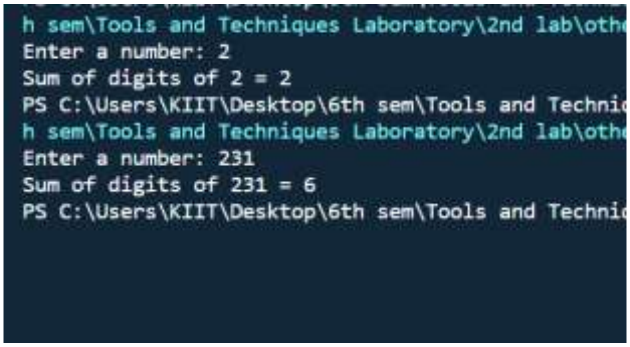
Program Title:

(WAP to calculate the sum of digits of a given number .)

Input/Output Screenshots:

RUN-1:

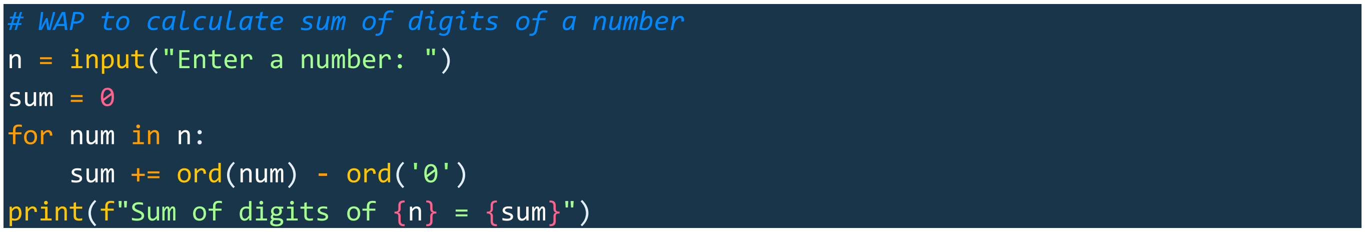
(Paste here the screenshots of first run)



```
h sem\Tools and Techniques Laboratory\2nd lab\othe
Enter a number: 2
Sum of digits of 2 = 2
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
h sem\Tools and Techniques Laboratory\2nd lab\othe
Enter a number: 231
Sum of digits of 231 = 6
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
```

Source code

(Paste here the source code)



```python
# WAP to calculate sum of digits of a number
n = input("Enter a number: ")
sum = 0
for num in n:
    sum += ord(num) - ord('0')
print(f"Sum of digits of {n} = {sum}")
```

Conclusion/Observation

Successfully compile to calculate the sum of digits of a given number

Program No: (3.)

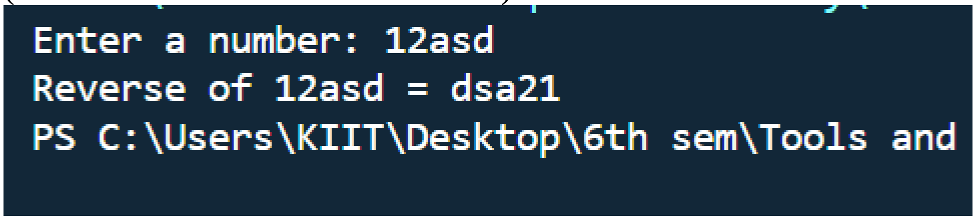
Program Title:

(WAP to display the reverse of a number entered through keyboard.)

Input/Output Screenshots:

RUN-1:

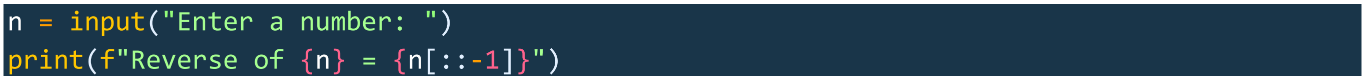
(Paste here the screenshots of first run)



```
Enter a number: 12asd
Reverse of 12asd = dsa21
PS C:\Users\KIIT\Desktop\6th sem\Tools and
```

Source code

(Paste here the source code)



```
n = input("Enter a number: ")
print(f"Reverse of {n} = {n[::-1]}")
```

Conclusion/Observation

Successfully compile to reverse the string

Program No: **(4.)**

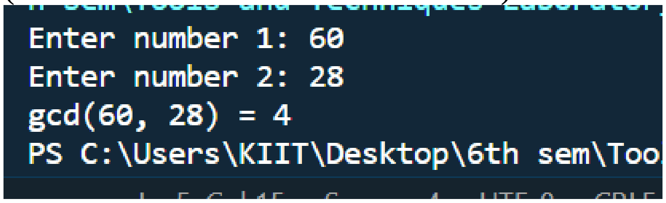
Program Title:

(WAP to find the GCD/HCF of two number.)

Input/Output Screenshots:

RUN-1:

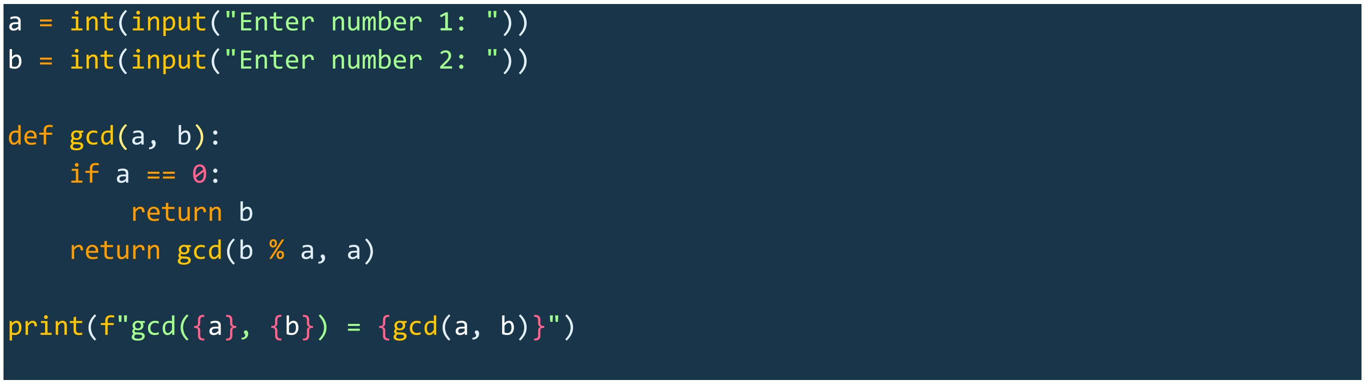
(Paste here the screenshots of first run)



```
Enter number 1: 60
Enter number 2: 28
gcd(60, 28) = 4
PS C:\Users\KIIT\Desktop\6th sem\Too
```

Source code

(Paste here the source code)



```
a = int(input("Enter number 1: "))
b = int(input("Enter number 2: "))

def gcd(a, b):
    if a == 0:
        return b
    return gcd(b % a, a)

print(f"gcd({a}, {b}) = {gcd(a, b)}")
```

Conclusion/Observation

Successfully compile to find gcd of two numbers

Program No: **(5.)**

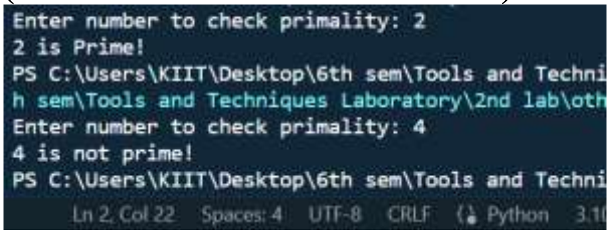
Program Title:

(# WAP to check if number is prime)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
Enter number to check primality: 2
2 is Prime!
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
h sem\Tools and Techniques Laboratory\2nd lab\oth
Enter number to check primality: 4
4 is not prime!
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
```

Source code

(Paste here the source code)

```
from math import sqrt

def isPrime(n):
    num = 2
    for num in range(2, int(sqrt(n)) + 1):
        if n % num == 0:
            return False
    return True

n = int(input("Enter number to check primality: "))
if isPrime(n):
    print(f"{n} is Prime!")
else:
    print(f"{n} is not prime!")
```

Conclusion/Observation

Successfully compile to check if number is prime or not

Program No: **(**.6)

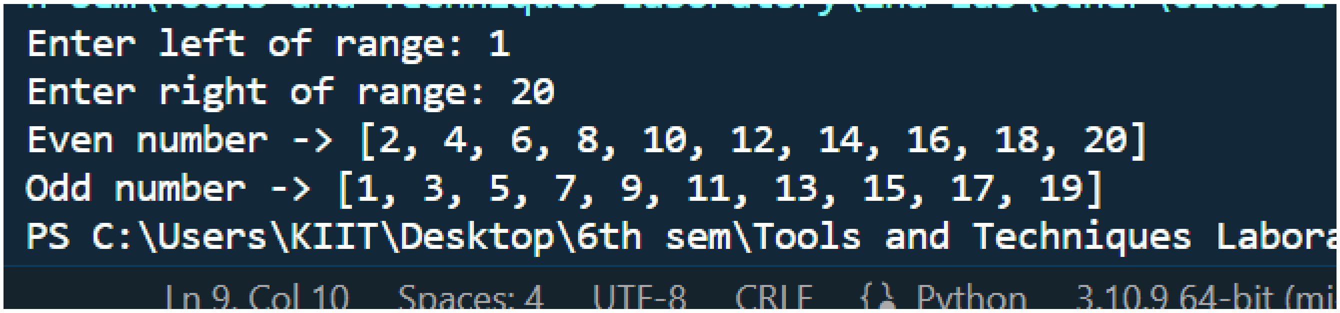
Program Title:

(# WAP to print all odd and even numbers separately within a given range. The range is input through user.)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
Enter left of range: 1
Enter right of range: 20
Even number -> [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Odd number -> [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Labora
```

Source code

(Paste here the source code)

```
# WAP to print all odd and even numbers separately within a given range. The range is
input through user.
l = int(input("Enter left of range: "))
r = int(input("Enter right of range: "))
odd = []
even = []
for num in range(l, r + 1):
    if num & 1:
        odd.append(num)
    else:
        even.append(num)

print(f"Even number -> {even}")
print(f"Odd number -> {odd}")
```

Conclusion/Observation

Successfully compile to print all odd and even numbers separately within a given range. The range is input through user.

Program No: (7.)

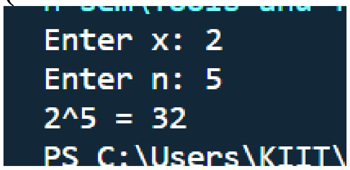
Program Title:

(# WAP to evaluate the equation y=x^n where n is a non-negative integer)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
Enter x: 2
Enter n: 5
2^5 = 32
PS C:\Users\KIIT\
```

Source code

(Paste here the source code)

```
x = int(input("Enter x: "))
n = int(input("Enter n: "))
print(f"{x}^{n} = {x**n}")
```

Conclusion/Observation

Successfully compile to evaluate the equation y=x^n where n is a non-negative integer.

Program No: (.8)

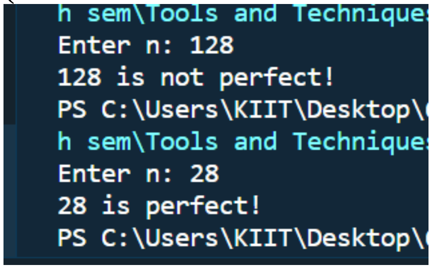
Program Title:

(#WAP to check whether an input integer is perfect number or not.)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
h sem\Tools and Technique
Enter n: 128
128 is not perfect!
PS C:\Users\KIIT\Desktop\
h sem\Tools and Technique
Enter n: 28
28 is perfect!
PS C:\Users\KIIT\Desktop\
```

Source code

(Paste here the source code)

```
# WAP to check whether an input integer is perfect number or not.
n = int(input("Enter n: "))

sum = 0
for num in range(1, n):
    if n % num == 0:
        sum += num

if sum == n:
    print(f"{n} is perfect!")
else:
    print(f"{n} is not perfect!")
```

Conclusion/Observation

Successfully compile to check whether an input integer is perfect number or not.

Program No: (9.)

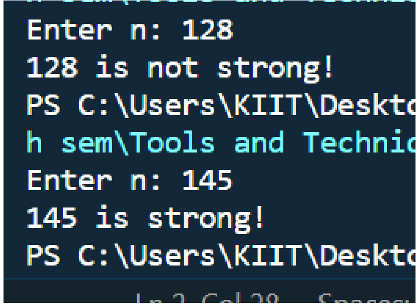
Program Title:

(#WAP to check whether an input integer is strong number or not.)

Input/Output Screenshots:

RUN-1:

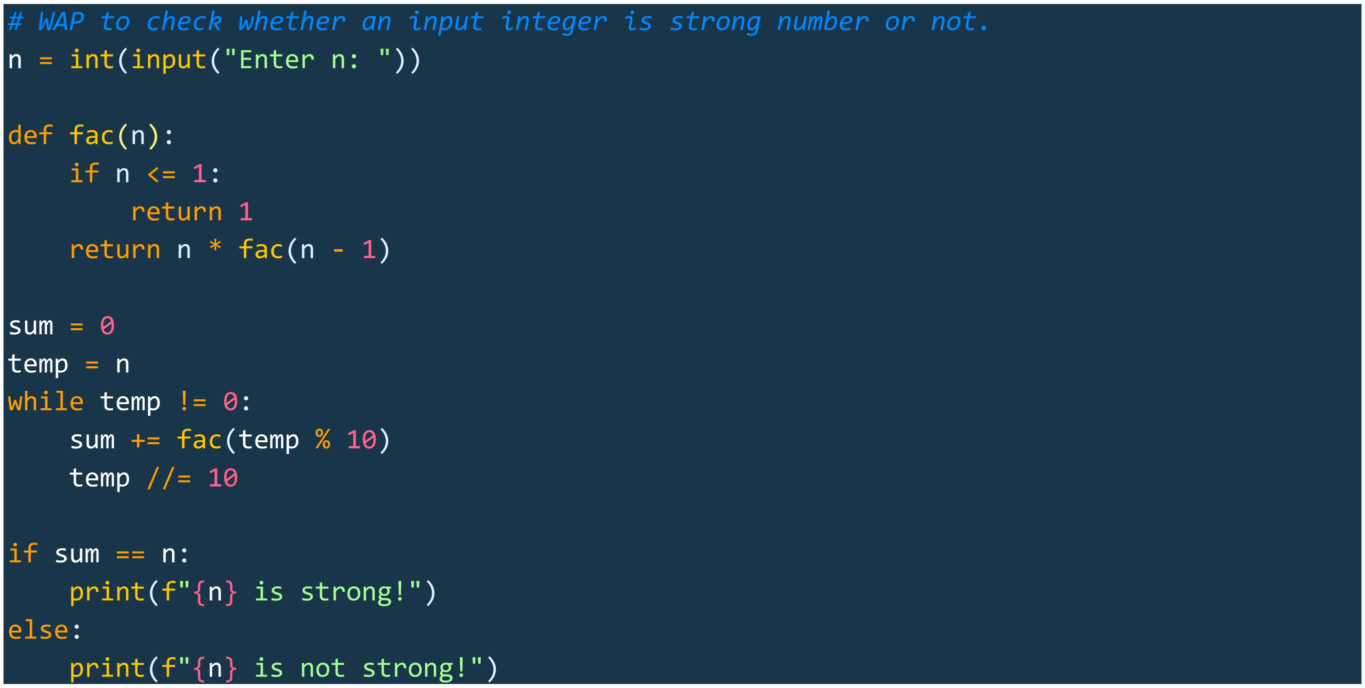
(Paste here the screenshots of first run)



```
Enter n: 128
128 is not strong!
PS C:\Users\KIIT\Deskto
h sem\Tools and Technic
Enter n: 145
145 is strong!
PS C:\Users\KIIT\Deskto
```

Source code

(Paste here the source code)



```python
# WAP to check whether an input integer is strong number or not.
n = int(input("Enter n: "))

def fac(n):
    if n <= 1:
        return 1
    return n * fac(n - 1)

sum = 0
temp = n
while temp != 0:
    sum += fac(temp % 10)
    temp //= 10

if sum == n:
    print(f"{n} is strong!")
else:
    print(f"{n} is not strong!")
```

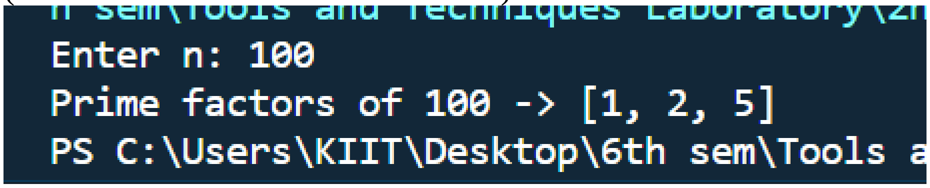
Conclusion/Observation

Successfully compile to check whether an input integer is strong number or not

Program No: **(10**.)

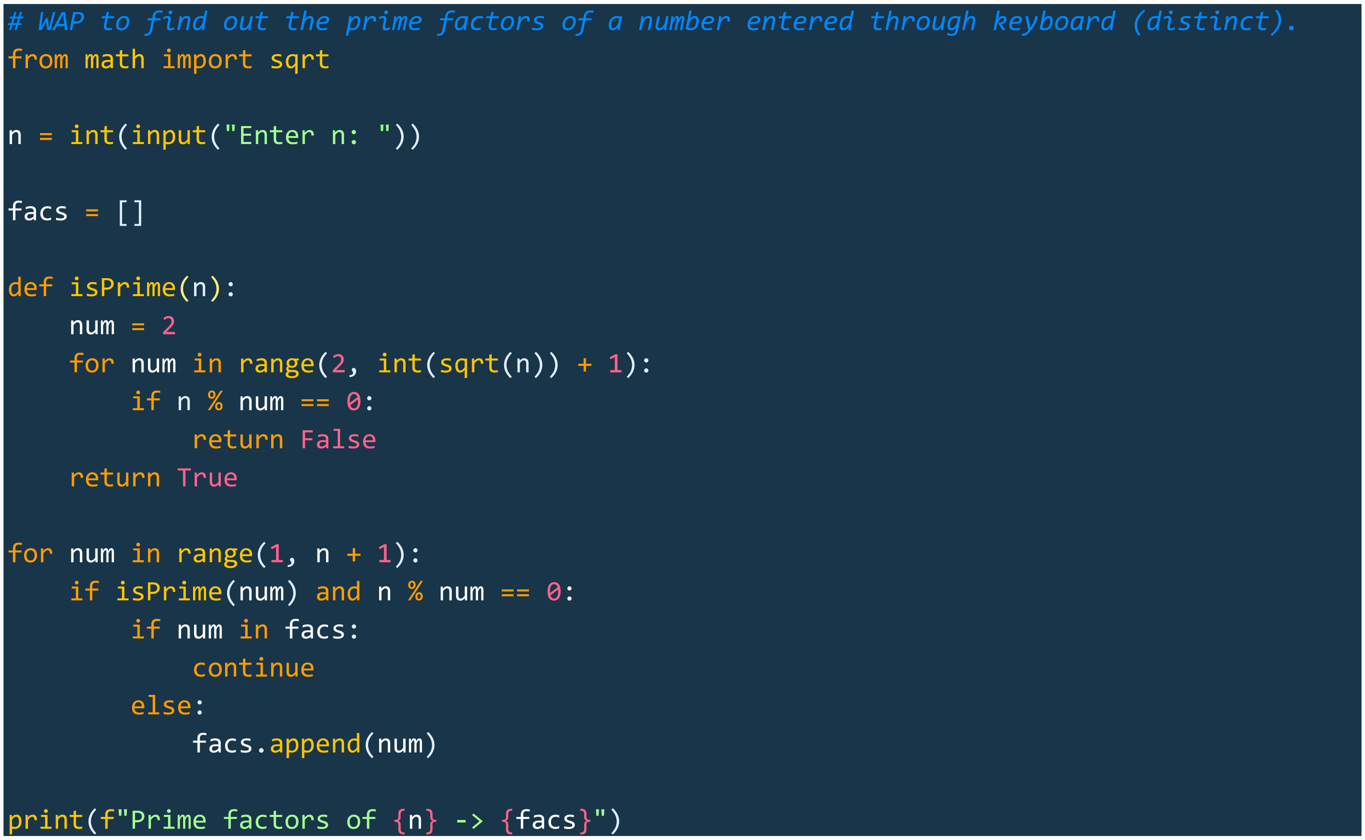
Program Title:
(# WAP to find out the prime factors of a number entered through keyboard (distinct).)

Input/Output Screenshots:
RUN-1:
(Paste here the screenshots of first run)



```
n sem\Tools and Techniques Laboratory\2n
Enter n: 100
Prime factors of 100 -> [1, 2, 5]
PS C:\Users\KIIT\Desktop\6th sem\Tools a
```

Source code
(Paste here the source code)



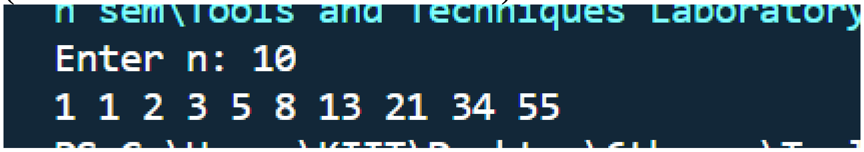
```python
# WAP to find out the prime factors of a number entered through keyboard (distinct).
from math import sqrt

n = int(input("Enter n: "))

facs = []

def isPrime(n):
    num = 2
    for num in range(2, int(sqrt(n)) + 1):
        if n % num == 0:
            return False
    return True

for num in range(1, n + 1):
    if isPrime(num) and n % num == 0:
        if num in facs:
            continue
        else:
            facs.append(num)

print(f"Prime factors of {n} -> {facs}")
```

Conclusion/Observation
Successfully compile to find out the prime factors of a number entered through keyboard (distinct

Program No: **(11**.)

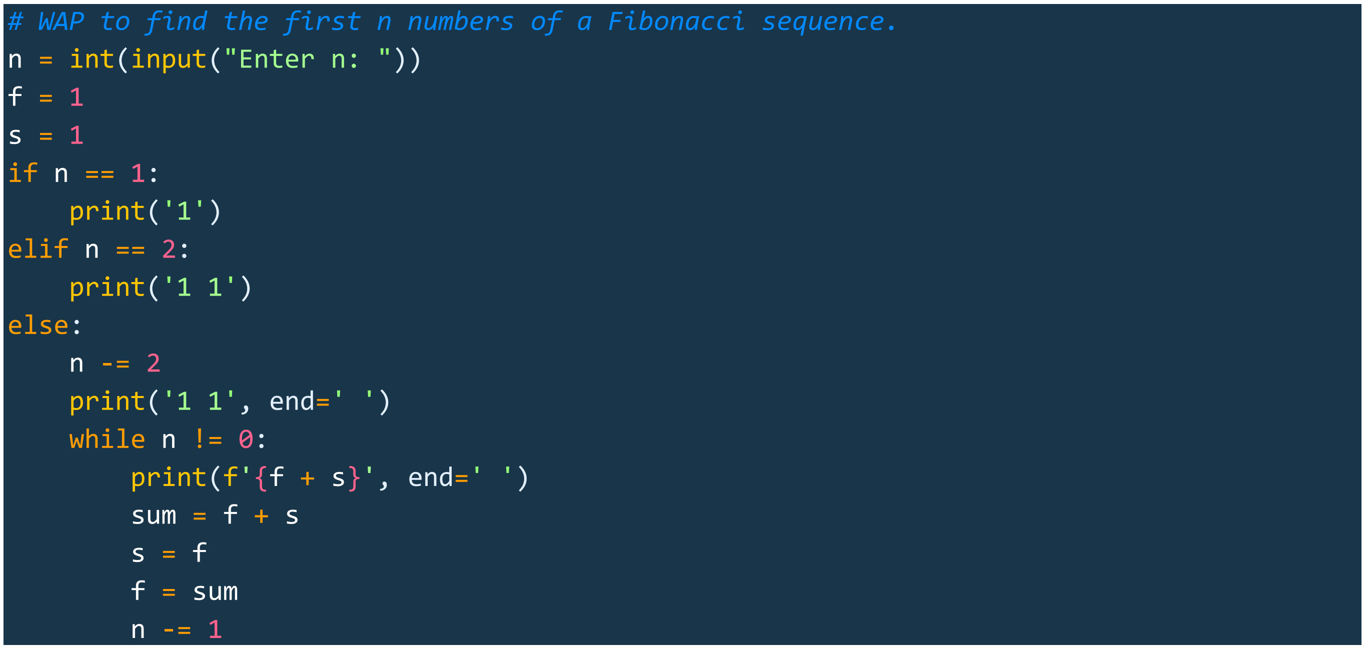
Program Title:
(# WAP to find the first n numbers of a Fibonacci sequence.)

Input/Output Screenshots:
RUN-1:
(Paste here the screenshots of first run)



```
n sem\Tools and Techniques Laboratory
Enter n: 10
1 1 2 3 5 8 13 21 34 55
```

Source code
(Paste here the source code)



```python
# WAP to find the first n numbers of a Fibonacci sequence.
n = int(input("Enter n: "))
f = 1
s = 1
if n == 1:
    print('1')
elif n == 2:
    print('1 1')
else:
    n -= 2
    print('1 1', end=' ')
    while n != 0:
        print(f'{f + s}', end=' ')
        sum = f + s
        s = f
        f = sum
        n -= 1
```

Conclusion/Observation
Successfully compile to find the first n numbers of a Fibonacci sequence.)

Program No: (12.)

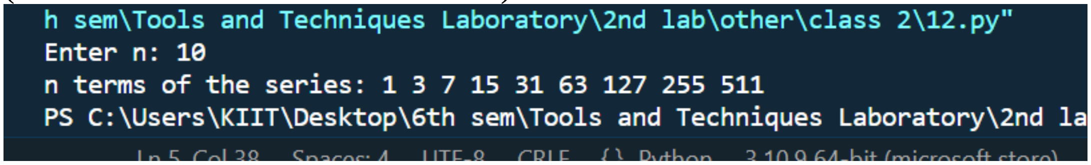
Program Title:

(# WAP to print the series as 1 3 7 15 31 ...., where n is given by user.)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)

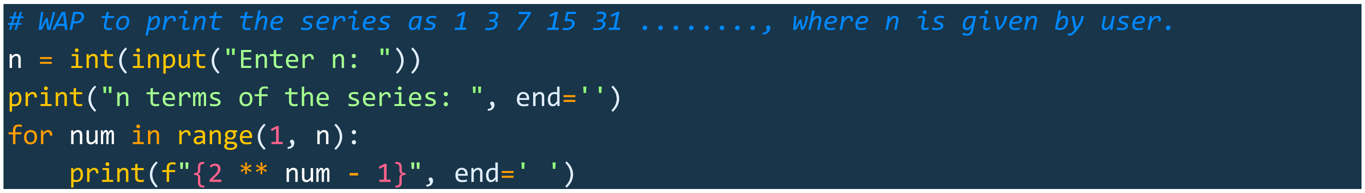


```
h sem\Tools and Techniques Laboratory\2nd lab\other\class 2\12.py"
Enter n: 10
n terms of the series: 1 3 7 15 31 63 127 255 511
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd la
```

Ln 5, Col 38   Spaces: 4   UTF-8   CRLF   {} Python   3.10.9 64-bit (microsoft store)

Source code

(Paste here the source code)



```python
# WAP to print the series as 1 3 7 15 31 ........, where n is given by user.
n = int(input("Enter n: "))
print("n terms of the series: ", end='')
for num in range(1, n):
    print(f"{2 ** num - 1}", end=' ')
```

Conclusion/Observation

Successfully compile to the series as 1 3 7 15 31 ...., where n is given by user.

Program No: (13.)

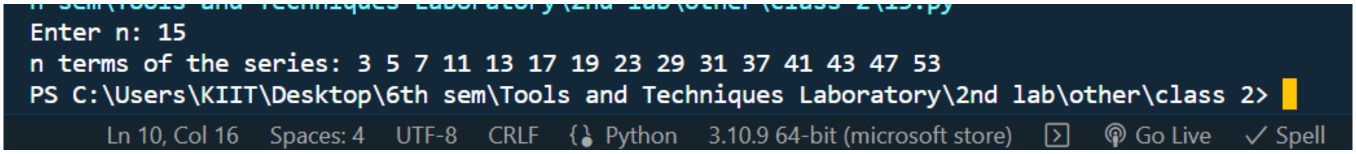
Program Title:

(# WAP to print the series as 3 5 7 11 13 17..........n, where n is given by user)

Input/Output Screenshots:

RUN-1:

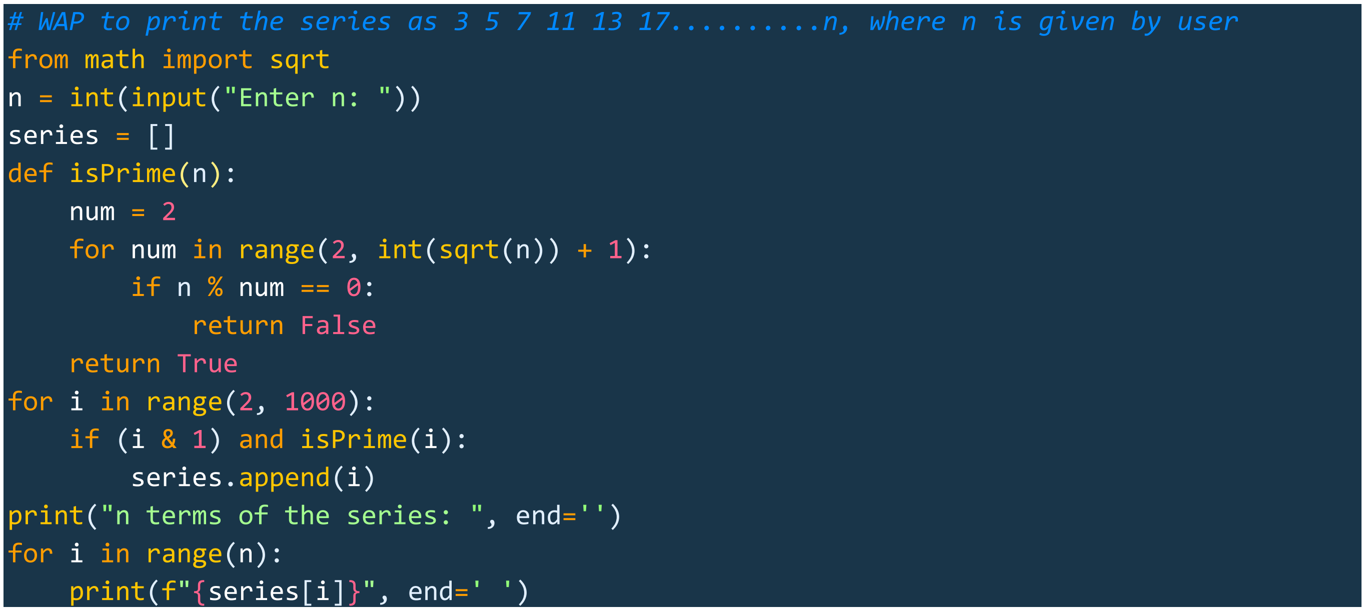
(Paste here the screenshots of first run)



```
h sem\Tools and Techniques Laboratory\2nd lab\other\class 2\13.py
Enter n: 15
n terms of the series: 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\class 2>
```

Source code

(Paste here the source code)



```python
# WAP to print the series as 3 5 7 11 13 17..........n, where n is given by user
from math import sqrt
n = int(input("Enter n: "))
series = []
def isPrime(n):
    num = 2
    for num in range(2, int(sqrt(n)) + 1):
        if n % num == 0:
            return False
    return True
for i in range(2, 1000):
    if (i & 1) and isPrime(i):
        series.append(i)
print("n terms of the series: ", end='')
for i in range(n):
    print(f"{series[i]}", end=' ')
```

Conclusion/Observation

Successfully compile to *the series as 3 5 7 11 13 17..........n, where n is given by user*

Program No: (14.)

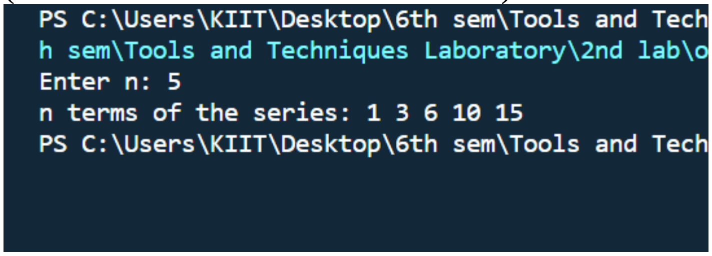
Program Title:

(#WAP to sum the following series S=1+(1+2)+(1+2+3)+...+(1+2+3+...+n))

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Tech
h sem\Tools and Techniques Laboratory\2nd lab\o
Enter n: 5
n terms of the series: 1 3 6 10 15
PS C:\Users\KIIT\Desktop\6th sem\Tools and Tech
```

Source code

(Paste here the source code)

```
# WAP to sum the following series S=1+(1+2)+(1+2+3)+...+(1+2+3+...+n)
n = int(input("Enter n: "))
print("n terms of the series: ", end='')
for num in range(1, n + 1):
    sum = 0
    for i in range(1, num + 1):
        sum += i
    print(f"{sum}", end=' ')
```

Conclusion/Observation

Successfully compile to sum the following series S=1+(1+2)+(1+2+3)+...+(1+2+3+...+n)

Program No: **(15**.)

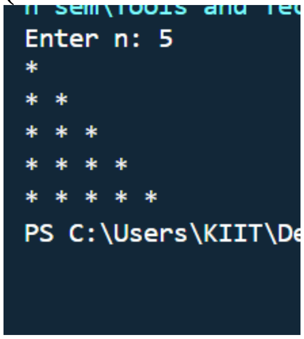
Program Title:

(# WAP to print the following pattern for n rows. Ex. for n=5 rows)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
n sem\Tools and Tec
Enter n: 5
*
* *
* * *
* * * *
* * * * *
PS C:\Users\KIIT\De
```

Source code

(Paste here the source code)

```
n = int(input("Enter n: "))
for i in range(1, n + 1):
    for j in range(1, i + 1):
        print('*', end=' ')
    print('')
```

Conclusion/Observation

Successfully compile to print the following pattern for n rows. Ex. for n=5 rows

Program No: (**16**.)

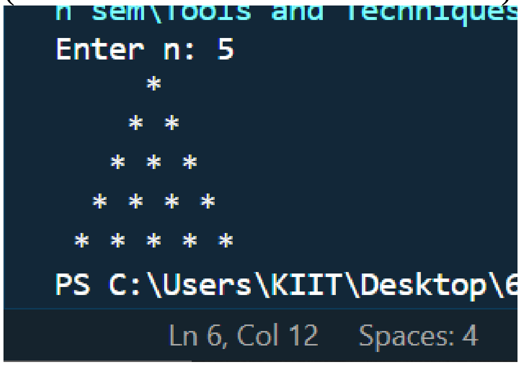
Program Title:

(# WAP to print the Following pattern for n rows. Ex. for n=5 rows)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)

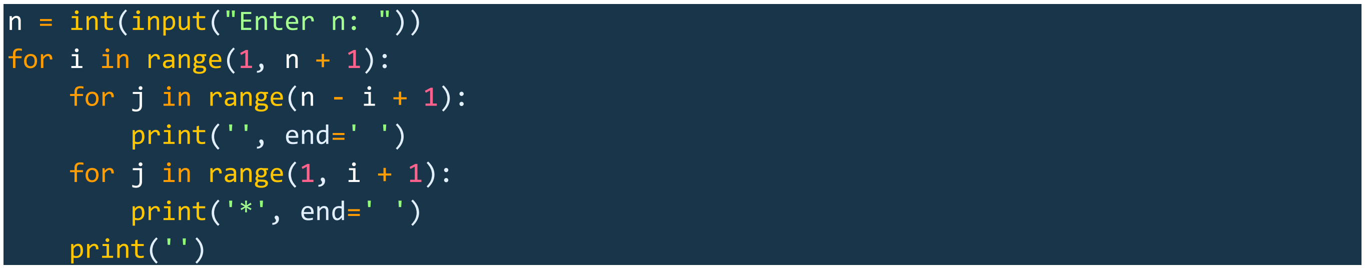


```
n sem\Tools and Techniques
Enter n: 5
     *
    * *
   * * *
  * * * *
 * * * * *
PS C:\Users\KIIT\Desktop\6
```

Ln 6, Col 12   Spaces: 4

Source code

(Paste here the source code)



```
n = int(input("Enter n: "))
for i in range(1, n + 1):
    for j in range(n - i + 1):
        print('', end=' ')
    for j in range(1, i + 1):
        print('*', end=' ')
    print('')
```

Conclusion/Observation

Successfully compile to print the Following pattern for n rows. Ex. for n=5 rows

Program No: (**17**.)

Program Title:

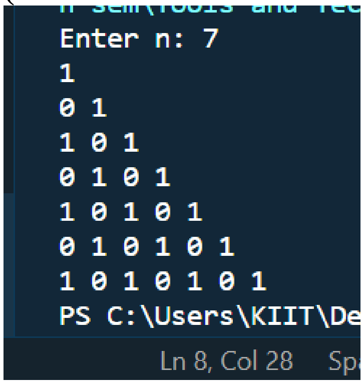
(# WAP to print the following pattern for n rows. Ex. for n=6 rows

```
# 1
# 0     1
# 1     0 1
# 0     1 0     1
# 1     0 1     0 1
# 0     1 0     1 0     1 #)
```

Input/Output Screenshots:

RUN-1:

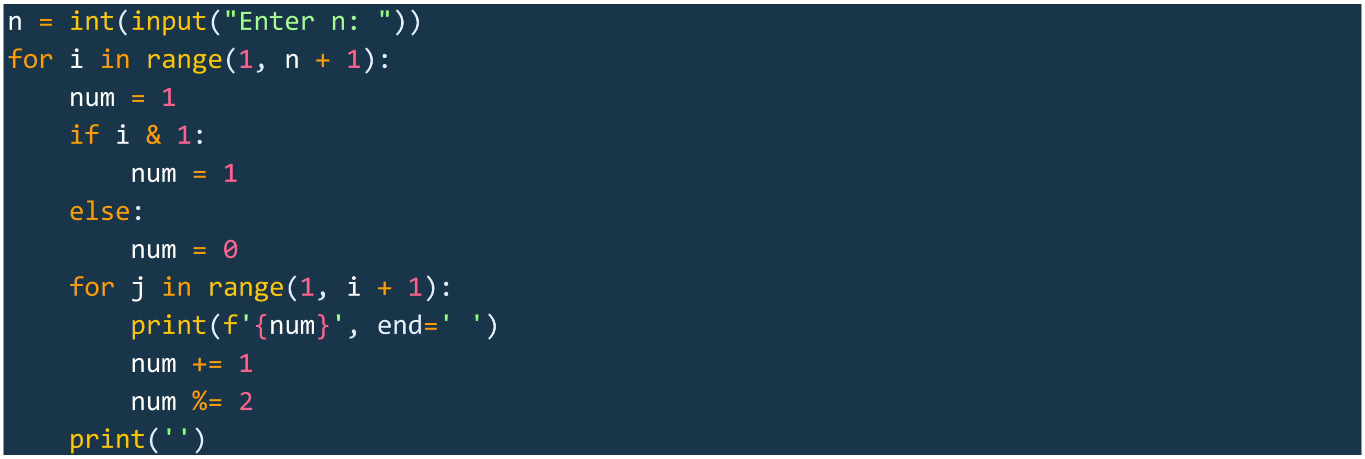
(Paste here the screenshots of first run)



```
Enter n: 7
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
1 0 1 0 1 0 1
PS C:\Users\KIIT\De
```

Source code

(Paste here the source code)



```python
n = int(input("Enter n: "))
for i in range(1, n + 1):
    num = 1
    if i & 1:
        num = 1
    else:
        num = 0
    for j in range(1, i + 1):
        print(f'{num}', end=' ')
        num += 1
        num %= 2
    print('')
```

Conclusion/Observation

Successfully compile to to print the following pattern for n rows. Ex. for n=6 rows

Program No: **(18.)**

Program Title:

(# # WAP to print the following pattern for n rows. Ex. for n=5 rows

A

B A

# C B A

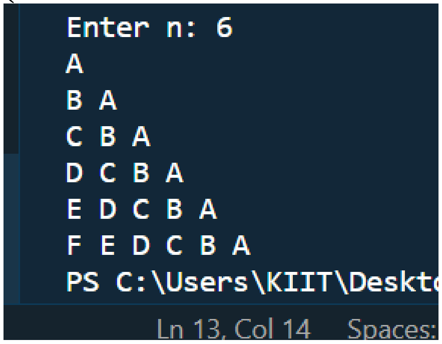
# D C B A

# E D C B A

Input/Output Screenshots:

RUN-1:

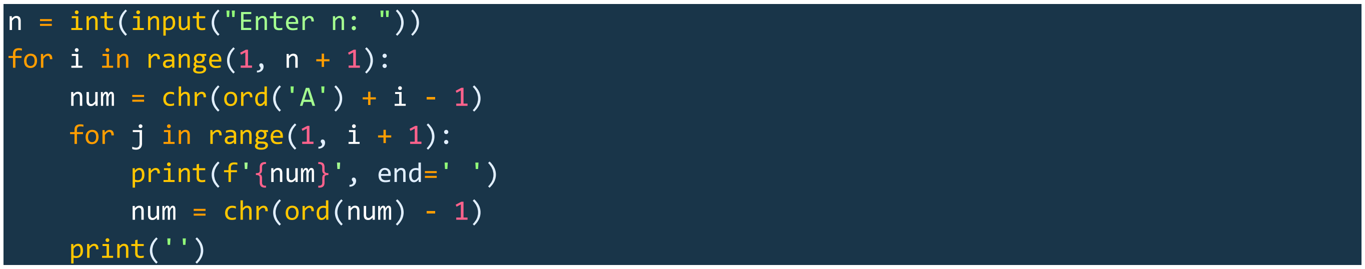
(Paste here the screenshots of first run)



```
Enter n: 6
A
B A
C B A
D C B A
E D C B A
F E D C B A
PS C:\Users\KIIT\Deskt
```

Source code

(Paste here the source code)



```python
n = int(input("Enter n: "))
for i in range(1, n + 1):
    num = chr(ord('A') + i - 1)
    for j in range(1, i + 1):
        print(f'{num}', end=' ')
        num = chr(ord(num) - 1)
    print('')
```

Conclusion/Observation

Successfully compile to print the following pattern for n rows. Ex. for n=5 rows

---

Program No: (19.)

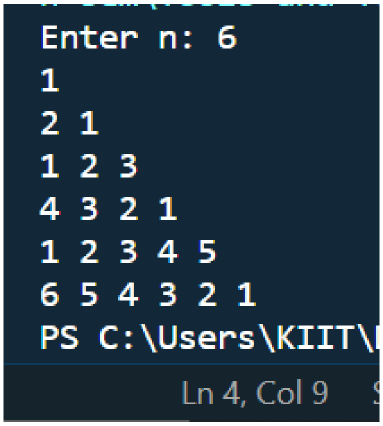
Program Title:

(## 19.WAP to print the following pattern for n rows. Ex. for n=5 rows
1
2 1
# 1 2 3
# 4 3 2 1
# 1 2 3 4 5)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
Enter n: 6
1
2 1
1 2 3
4 3 2 1
1 2 3 4 5
6 5 4 3 2 1
PS C:\Users\KIIT\
```

Source code

(Paste here the source code)

```
n = int(input("Enter n: "))
for i in range(1, n + 1):
    if i & 1 == 0:
        num = i
        for j in range(1, i + 1):
            print(f'{num}', end=' ')
            num -= 1
    else:
        num = 1
        for j in range(1, i + 1):
            print(f'{num}', end=' ')
            num += 1
    print('')
```

Conclusion/Observation

Successfully compile to print the following pattern for n rows. Ex. for n=5 rows

Program No: (20.)

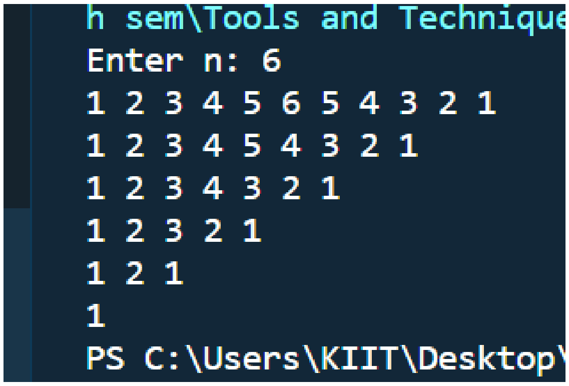
Program Title:

(## # WAP to form reverse pyramid of numbers for a given number Ex: n = 4
# 1 2 3 4 3 2 1
# 1 2 3 2 1
# 1 2 1
1)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
h sem\Tools and Technique
Enter n: 6
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1
PS C:\Users\KIIT\Desktop\
```

Source code

(Paste here the source code)

```
n = int(input("Enter n: "))
for i in range(1, n + 1):
    for j in range(1, n - i + 2):
        print(f'{j}', end=' ')
    j = n - i
    while j > 0:
        print(f'{j}', end=' ')
        j -= 1
    print('')
```

Conclusion/Observation

Successfully compile to form reverse pyramid of numbers for a given number Ex: n = 4

Program No: (21.)

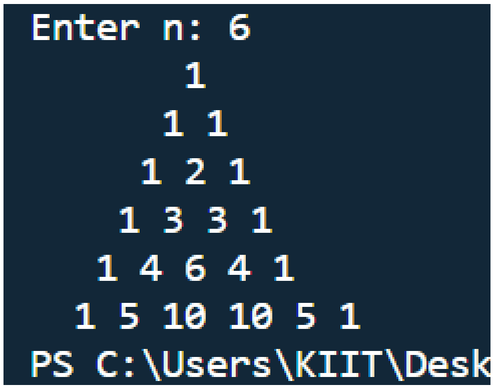
Program Title:

(## # WAP to generate the pascal triangle pyramid of numbers for a given number. Ex. for number 4 1
1 1
# 1 2 1
# 1 3 3 1
# 1 4 6 4 1

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)



```
 Enter n: 6
        1
       1 1
      1 2 1
     1 3 3 1
    1 4 6 4 1
   1 5 10 10 5 1
 PS C:\Users\KIIT\Desk
```

Source code

(Paste here the source code)

```
from math import factorial

n = int(input("Enter n: "))
for i in range(n):
    for j in range(n-i+1):
        print(end=" ")
    for j in range(i+1):
        print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")
    print()
```

Conclusion/Observation

Successfully compile to generate the pascal triangle pyramid of numbers for a given number. Ex. for number 4 1

Program No: **(22.)**

Program Title:

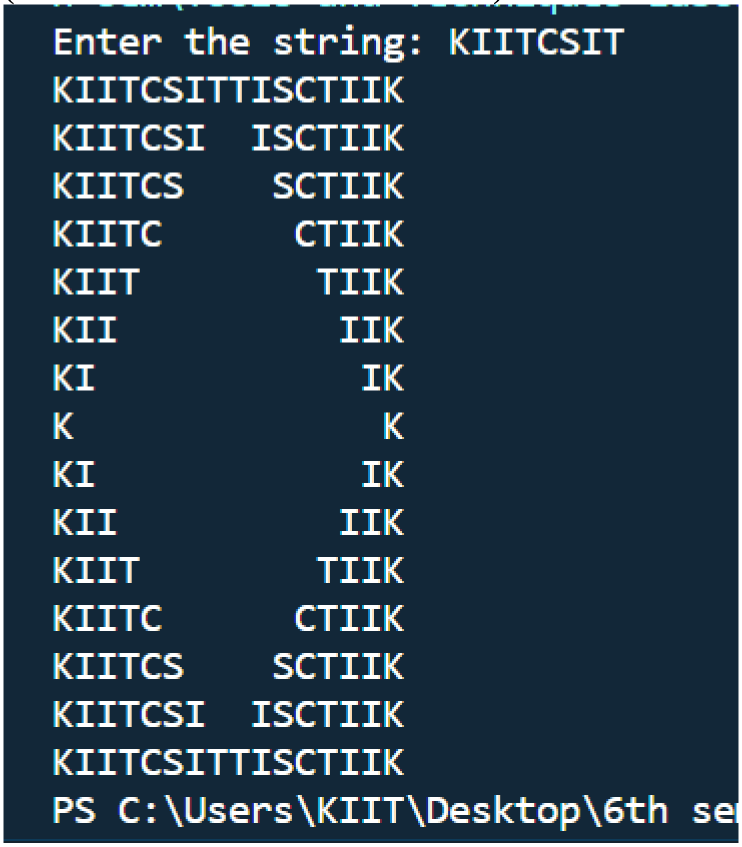
(## WAP to display the following style o/p for a given string input through keyboard.

```
# (Ex.for a string "KIITCSIT")
# KIITCSITTISCTIIK
# KIITCSI  ISCTIIK
# KIITCS    SCTIIK
# KIITC      CTIIK
# KIIT        TIIK
# KII          IIK
# KI            IK
# K              K
# KI            IK
# KII          IIK
# KIIT        TIIK
# KIITC      CTIIK
# KIITCS        SCTIIK
# KIITCSI  ISCTIIK
# KIITCSITTISCTIIK
```

Input/Output Screenshots:

RUN-1:

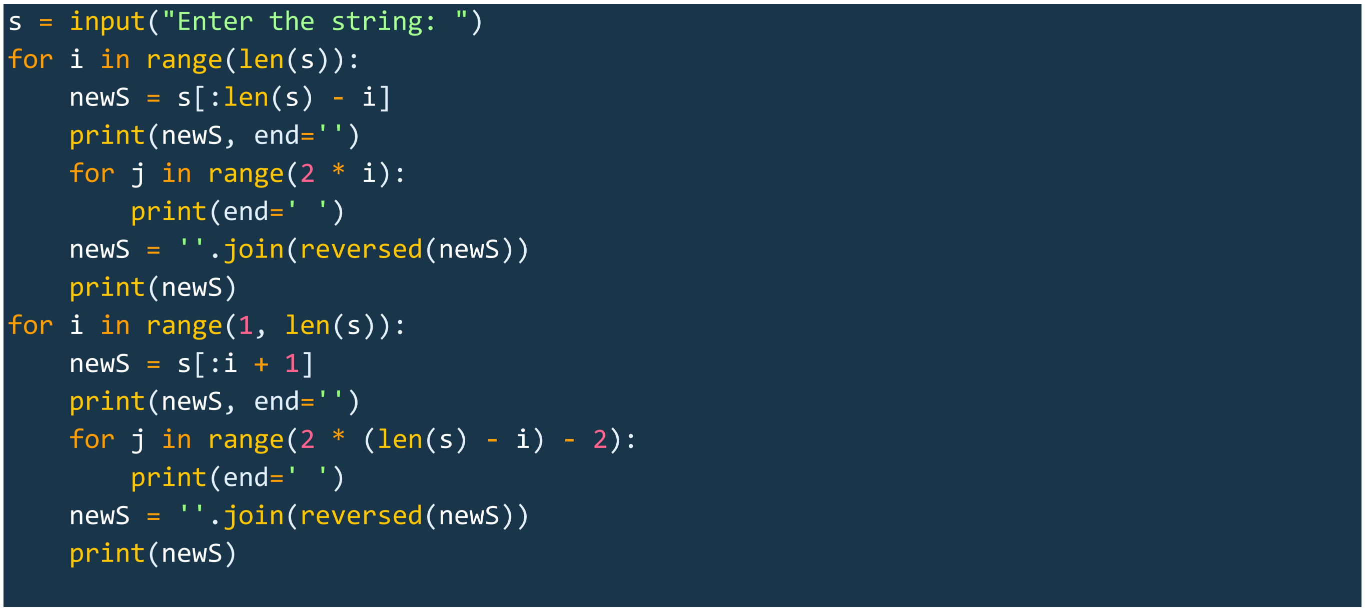
(Paste here the screenshots of first run)



```
Enter the string: KIITCSIT
KIITCSITTISCTIIK
KIITCSI  ISCTIIK
KIITCS    SCTIIK
KIITC      CTIIK
KIIT        TIIK
KII          IIK
KI            IK
K              K
KI            IK
KII          IIK
KIIT        TIIK
KIITC      CTIIK
KIITCS    SCTIIK
KIITCSI  ISCTIIK
KIITCSITTISCTIIK
PS C:\Users\KIIT\Desktop\6th se
```

Source code

(Paste here the source code)



```python
s = input("Enter the string: ")
for i in range(len(s)):
    newS = s[:len(s) - i]
    print(newS, end='')
    for j in range(2 * i):
        print(end=' ')
    newS = ''.join(reversed(newS))
    print(newS)
for i in range(1, len(s)):
    newS = s[:i + 1]
    print(newS, end='')
    for j in range(2 * (len(s) - i) - 2):
        print(end=' ')
    newS = ''.join(reversed(newS))
    print(newS)
```

Conclusion/Observation

Successfully compile to display the following style o/p for a given string input through keyboard.
# (Ex.for a string "KIITCSIT")

Program No: (23.)

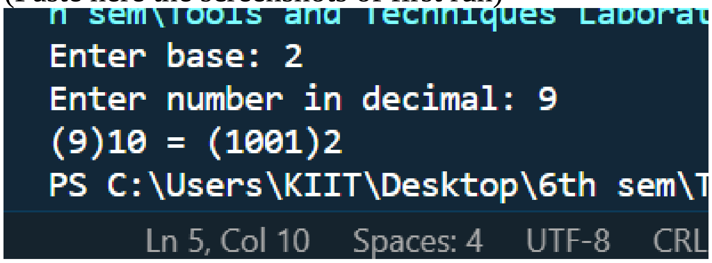
Program Title:

(## # WAP WAP to convert a decimal number into its equivalent number with base b. Decimal number and b are the user input
# Assuming 2 <= b <= 9

Input/Output Screenshots:

RUN-1:

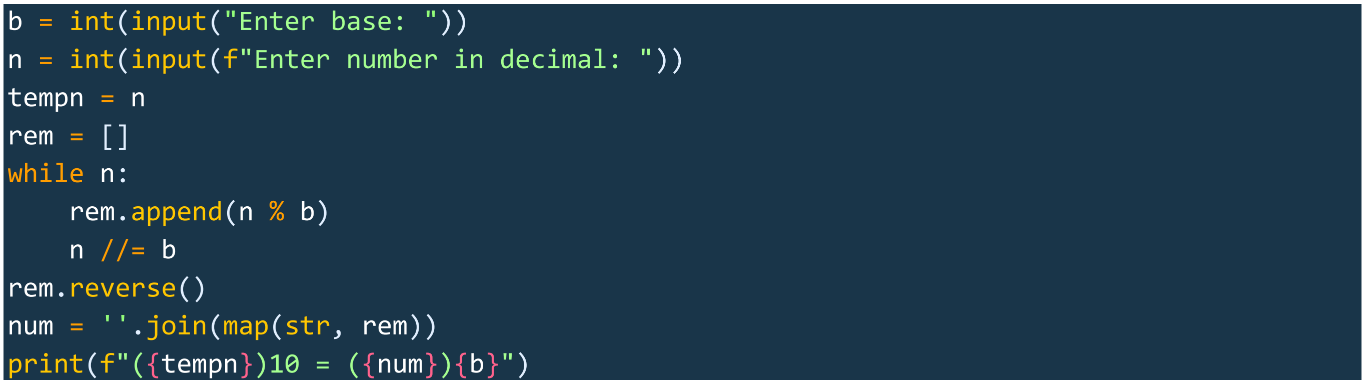
(Paste here the screenshots of first run)



```
n sem\Tools and Techniques Laborat
Enter base: 2
Enter number in decimal: 9
(9)10 = (1001)2
PS C:\Users\KIIT\Desktop\6th sem\T
```

Source code

(Paste here the source code)



```python
b = int(input("Enter base: "))
n = int(input(f"Enter number in decimal: "))
tempn = n
rem = []
while n:
    rem.append(n % b)
    n //= b
rem.reverse()
num = ''.join(map(str, rem))
print(f"({tempn})10 = ({num}){b}")
```

Conclusion/Observation

Successfully compile to convert a decimal number into its equivalent number with base b. Decimal number and b are the user input
# Assuming 2 <= b <= 9

Program No: **(24.)**

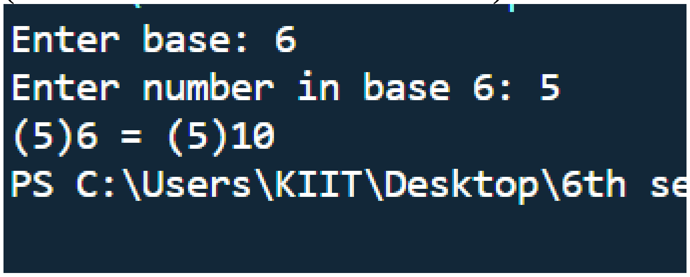
Program Title:

(## # WAP to to convert a number with base b into its equivalent decimal number. Number with base b & b are the user input.)

Input/Output Screenshots:

RUN-1:

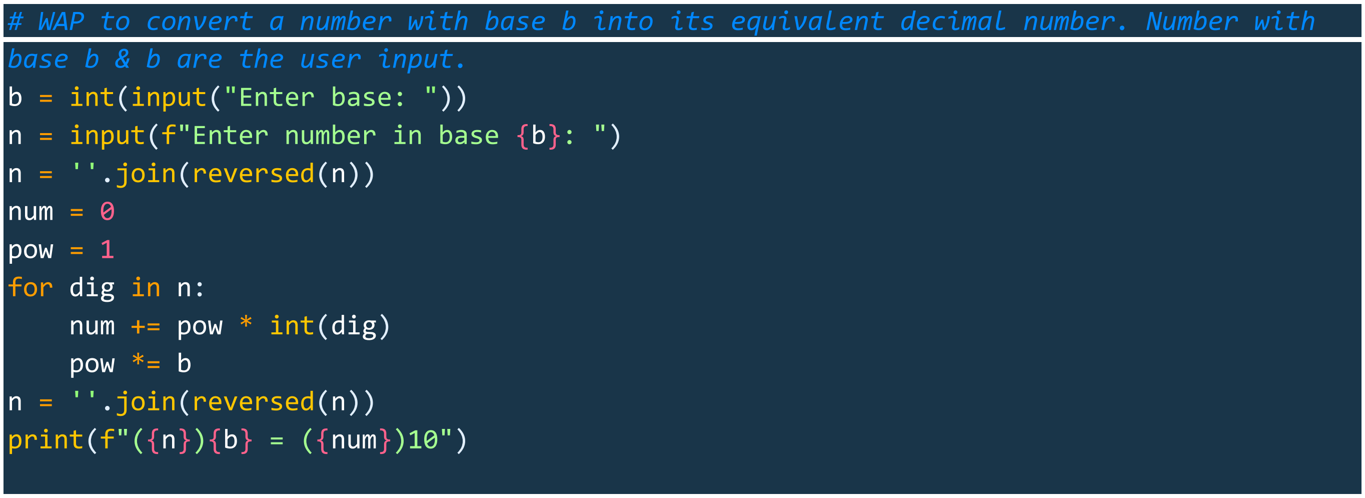
(Paste here the screenshots of first run)



```
Enter base: 6
Enter number in base 6: 5
(5)6 = (5)10
PS C:\Users\KIIT\Desktop\6th se
```

Source code

(Paste here the source code)



```
# WAP to convert a number with base b into its equivalent decimal number. Number with
base b & b are the user input.
b = int(input("Enter base: "))
n = input(f"Enter number in base {b}: ")
n = ''.join(reversed(n))
num = 0
pow = 1
for dig in n:
    num += pow * int(dig)
    pow *= b
n = ''.join(reversed(n))
print(f"({n}){b} = ({num})10")
```

Conclusion/Observation

Successfully compile to convert a number with base b into its equivalent decimal number. Number with base b & b are the user input.

Program No: **(20.)**

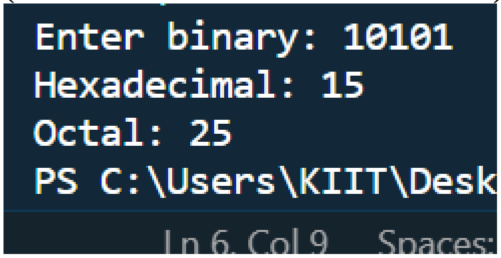
Program Title:

(## # WAP To convert a binary number to its equivalent octal & hexa-decimal number system.# 1 2 1

Input/Output Screenshots:

RUN-1:

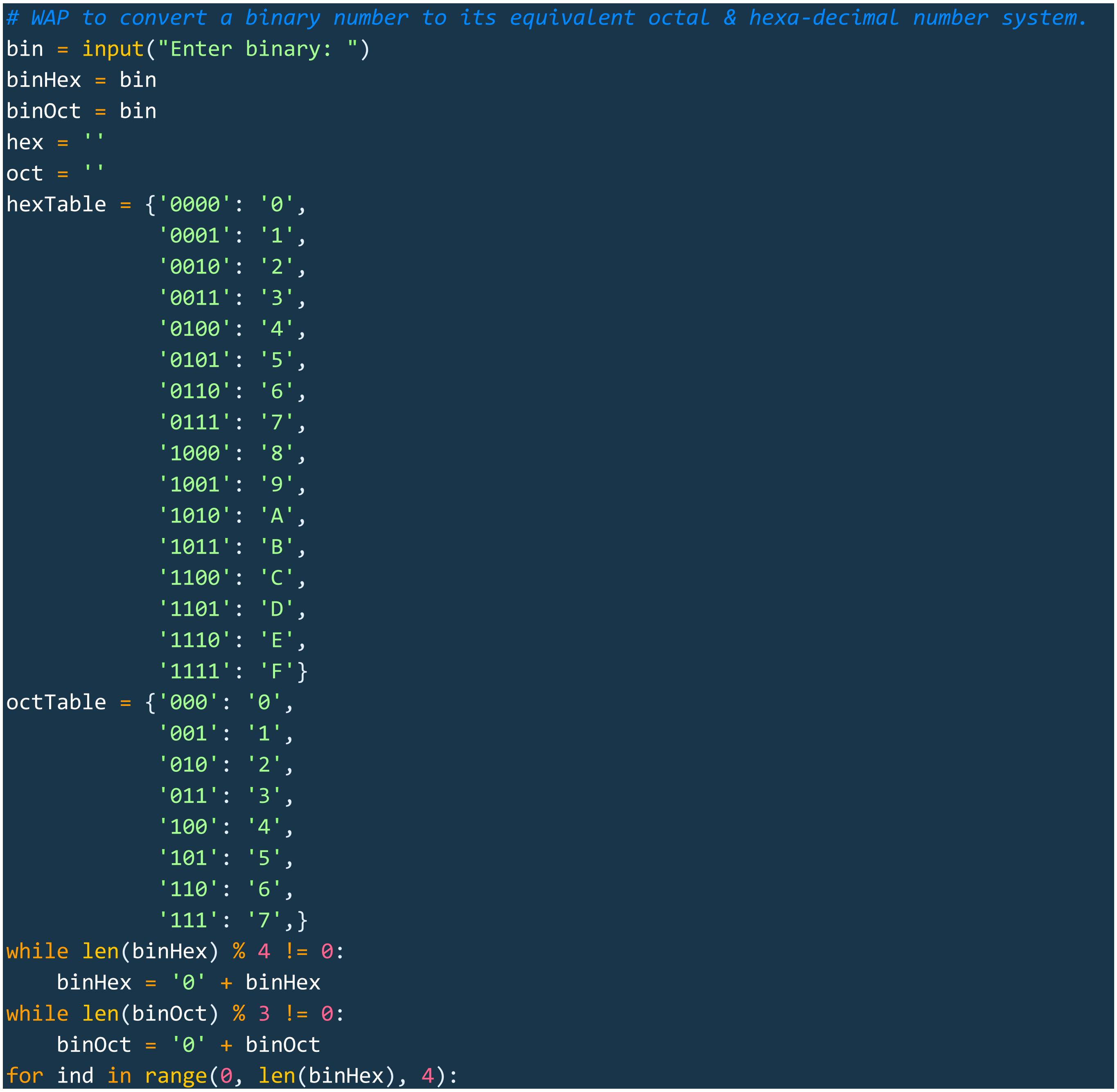
(Paste here the screenshots of first run)



```
Enter binary: 10101
Hexadecimal: 15
Octal: 25
PS C:\Users\KIIT\Desk
```

Source code

(Paste here the source code)



```python
# WAP to convert a binary number to its equivalent octal & hexa-decimal number system.
bin = input("Enter binary: ")
binHex = bin
binOct = bin
hex = ''
oct = ''
hexTable = {'0000': '0',
            '0001': '1',
            '0010': '2',
            '0011': '3',
            '0100': '4',
            '0101': '5',
            '0110': '6',
            '0111': '7',
            '1000': '8',
            '1001': '9',
            '1010': 'A',
            '1011': 'B',
            '1100': 'C',
            '1101': 'D',
            '1110': 'E',
            '1111': 'F'}
octTable = {'000': '0',
            '001': '1',
            '010': '2',
            '011': '3',
            '100': '4',
            '101': '5',
            '110': '6',
            '111': '7',}
while len(binHex) % 4 != 0:
    binHex = '0' + binHex
while len(binOct) % 3 != 0:
    binOct = '0' + binOct
for ind in range(0, len(binHex), 4):
    conv = binHex[ind] + binHex[ind + 1] + binHex[ind + 2] + binHex[ind + 3]
    hex += hexTable[conv]
for ind in range(0, len(binOct), 3):
    conv = binOct[ind] + binOct[ind + 1] + binOct[ind + 2]
    oct += octTable[conv]

print(f"Hexadecimal: {hex}")
print(f"Octal: {oct}")
```

Conclusion/Observation

Successfully compile to convert a binary number to its equivalent octal & hexa-decimal number system.# 1 2 1