



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)

```
# WAP to calculate the factorial of a given by function call
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n - 1)
n = int(input("Enter number to calculate factorial: "))
print(f"Factorial of {n} is {factorial(n)}")
```

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)

```
# WAP to calculate sum of digits of a number by function
def sumOfDigits(n):
    if n == 0:
        return 0
    return n % 10 + sumOfDigits(n // 10)
n = int(input("Enter number to calculate sum of digits: "))
print(f"Sum of digits of {n} is {sumOfDigits(n)}")
```

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)

```
# WAP to display reverse of a number by function call
def reverse(n):
    if n == 0:
        return 0
    return (n % 10) * 10 ** (len(str(n)) - 1) + reverse(n // 10)
n = int(input("Enter number to reverse: "))
print(f"Reverse of {n} is {reverse(n)}")
```

Conclusion/Observation

Successfully compile to reverse the string

Program No: (4.)

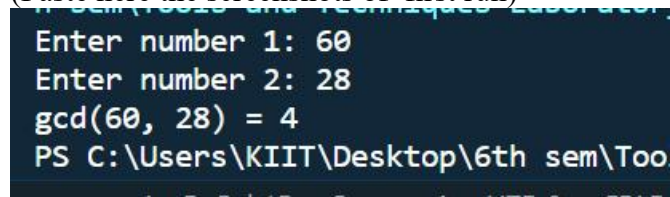
Program Title:

(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

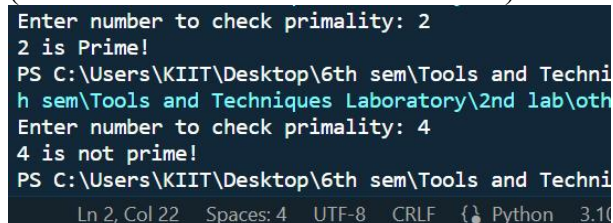
Program No: (5.)

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 Techniques Laboratory\2nd lab\other
Enter number to check primality: 4
4 is not prime!
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other
Ln 2, Col 22  Spaces: 4  UTF-8  CRLF  { } Python 3.10

```

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 find out the prime factors of a number entered through keyboard (distinct). /*Hints: A prime number is any number with no divisors other than itself and 1, such as 2 and 5. Any number can be written as a product of prime numbers in a unique way (except for the order). These are called prime factors of a number. In other words, In number theory, the prime factors of a positive integer are the prime numbers that divide that integer exactly, without leaving a remainder. The process of finding these numbers is called integer factorization, or prime factorization. • Enter a number : 100 • The prime factors of 100 are 2(2) and 5(2) • That is, $100 = 2 \times 2 \times 5 \times 5$, and those numbers are primes.)

Input/Output Screenshots:**RUN-1:**

(Paste here the screenshots of first run)

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\6.py"
Enter number to find prime factors: 45
3
3
5
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\6.py"
Spaces: 4 UTF-8 CRLF { } Python 3.10.9 64-bit
```

Source code

(Paste here the source code)

```
def primeFactors(n):
    num = 2
    while num <= n:
        if n % num == 0:
            print(num)
            n = n / num
        else:
            num = num + 1
n = int(input("Enter number to find prime factors: "))
primeFactors(n)
```

Conclusion/Observation

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

Program No: (7.)**Program Title:**

(# WAP to WAP to print the following pattern for n rows. Ex. for n=6 rows by function definition)

Input/Output Screenshots:**RUN-1:**

(Paste here the screenshots of first run)

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\7.py"
Enter n: 6
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\7.py"
Ln 19, Col 21 Spaces: 4 UTF-8 CRLF { } Python
```

Source code

(Paste here the source code)

```
def printPattern(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('')
n = int(input("Enter n: "))
printPattern(n)
```

Conclusion/Observation

Successfully compile to WAP to print the following pattern for n rows. Ex. for n=6 rows by function definition

Program No: (.8)**Program Title:**

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

Input/Output Screenshots:**RUN-1:**

(Paste here the screenshots of first run)

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\8.py"
Enter n: 7
A
B A
C B A
D C B A
E D C B A
F E D C B A
G F E D C B A
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\2nd lab\other\lab 3\8.py"
```

Source code

(Paste here the source code)

```
# WAP to print the following pattern for n rows. Ex. for n=5 rows by function definition
# A
# B A
# C B A
# D C B A
# E D C B A
def printPattern(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('\n')
n = int(input("Enter n: "))
printPattern(n)
```

Conclusion/Observation

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

Program No: (9.)

Program Title:

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

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques>
es Laboratory\2nd lab\other\lab 3\9.py"
Enter n: 7
1
2 1
1 2 3
4 3 2 1
1 2 3 4 5
6 5 4 3 2 1
1 2 3 4 5 6 7
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques>
```

Source code

(Paste here the source code)

```
# 19.WAP to print the following pattern for n rows. Ex. for n=5 rows by function definition
# 1
# 2 1
# 1 2 3
# 4 3 2 1
# 1 2 3 4 5
def printPattern(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
            print('\n')
        else:
            num = 1
            for j in range(1, i + 1):
                print(f'{num}', end=' ')
                num += 1
            print('\n')
```

```

    print('')
n = int(input("Enter n: "))
printPattern(n)

```

Conclusion/Observation

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

Program No: (10.)

Program Title:

(# WAP to # WAP to convert a decimal number into its equivalent number with base b. Decimal number and b are the user input using function deference
Assuming $2 \leq b \leq 9$.)

Input/Output Screenshots:

RUN-1:

(Paste here the screenshots of first run)

```

es Laboratory\2nd lab\other\lab 3\10.py
Enter base: 8
Enter number in decimal: 80
(80)10 = (120)8
PS C:\Users\KIIT\Desktop\6th sem\Tools
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF { }

```

Source code

(Paste here the source code)

```

# WAP to convert a decimal number into its equivalent number with base b. Decimal number and b are the user input using function
# deference
# Assuming 2 <= b <= 9
def convert(n, b):
    rem = []
    while n:
        rem.append(n % b)
        n //= b
    rem.reverse()
    num = ''.join(map(str, rem))
    return num
b = int(input("Enter base: "))
n = int(input(f"Enter number in decimal: "))
tempn = n
print(f"({tempn})10 = ({convert(n, b)}){b}")

```

Conclusion/Observation

Successfully compile to # WAP to convert a decimal number into its equivalent number with base b.

Decimal number and b are the user input using function deference

Assuming $2 \leq b \leq 9$

