NAME: Harsh Shah

Batch: C-31

Roll No.: 2003154

EXPERIMENT NO. 1

Harsh Shal C-31

2003154 __/__/____

Exp - 01

A) Aim: WAP in python to print following paragraph:

Theory: print () function:

- The python print () function ~~fund~~ takes in any number of arguments and prints them in a single line with one blankspace between each parameters.

- Though it is not necessary to pass arguments in the print () function, it requires an empty parenthesis at the end that tells python to execute the function rather calling it by name.

Syntax :- print ("Hello, World!")
output :- Hello, World!

- String literals :-

\n :- This string literal is used to add a new blank line while printing a statement.
\t :- This string literal is used to add a tab space in the current line.

- In the interpreter, standard output displays to the screen in between the ">>>" prompts, so its an easy way to see what these \n print() function.

Program:

```
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.\nTwinkle, ' twinkle', little star,\n\tHow I wonder what you are")
```

Output:

```
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1a.py
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
```

Harsh Shah C31
2003154

B] Aim :- WAP to show output formatting take two values and display them using single print function using

- str.format()
- % operator.

Theory :- str() function :-

The str() function returns the string version of the given object.

The str() method takes three parameters :-

- object - The object whose string representation is to be returned. If not provided, returns the empty string.
- encoding - Encoding of the given object. Default is of UTF-8 when not provided.
- errors - Response when decoding fails.

str.format() - also known as 'f' strings now allow us to include variables directly into string. syntax :- f"Hello, ${Name}"
output :- If name = Harsh then => Hello, Harsh.

% operator :- It is used to find the remainder after dividing two numbers.

Program:

```
first = input("Enter the first number: ")
second = input("Enter the second number: ")
print("The first value is {} and second value is %d".format(first) % int(second))
```

Output:

```
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1b.py
Enter the first number: 10
Enter the second number: 21
The first value is 10 and second value is 21
```

Harsh Shah C31

2003114

__/__/____

C] Aim :- WAP to find leap year using nested if

Theory :- Nested if in ~~Python~~ Python:

- There maybe a ~~st~~ situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if condition. In a nested if construct, you can have an if, elif, else construct inside another if, elif, else condition.

- Syntax :-

```
if exp1:
    statement (s)
elif exp 2:
    if exp 3:
        statement (s)
    else:
        statement(s)
else:
    statement (s)
```

Therefore the nested if construct help us to execute a set of instructions acc. to condition that is true in python.

Program:

```
year = int(input("Enter the year: "))
if(year%4==0):
 if(year%100==0):
  if(year%400==0):
   print("{} is a leap year".format(year))
  else:
   print("{} is not a leap year".format(year))
 else:
  print("{} is a leap year".format(year))
else:
 print("{} is not a leap year".format(year))
```

Output:

```
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1c.py
Enter the year: 2024
2024 is a leap year
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1c.py
Enter the year: 1985
1985 is not a leap year
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1c.py
Enter the year: 1800
1800 is not a leap year
```

Hansh Shah C31

2207154

__/__/____

D1 Aim: WAP to print Armstrong number in range 1 to 1000.

Theory:

- Armstrong Number: An armstrong number is equal to the cubes of its own digits,

for example: $153 = (1)^3 + (5)^3 + (3)^3$

- Loops in python: In this program we will use for loop.

- FOR loop:- For loops are used for sequential traverse.
It is used when the upper limit of the loop is known to the programmer.

Eg, Traversing a list or string or array, etc.

There is a for in a loop which is equal to for a loop in other programming languages.

Syntax:- iterator_var in sequence:
statement(s)

Program:

```
for i in range(1000):
 temp = i
 comp = 0
 while i:
  comp += (i % 10) ** 3
  i //= 10
 if temp == comp:
  print(temp)
```

Output:

```
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1d.py
1
153
370
371
407
```

Harsh Shah C31
2603154

__/__/____

E7 Aim :- WAP to find fibonacci series of n terms.

Theory : Fibonacci series :- In this sequence, the next number is the sum of two ~~preceding~~ preceding ones. It usually starts with 0 and 1.

Functions :- Python functions is a ~~bloo~~ block of related statements designed to perform a computational, logical, or evaluative task.

- Here, we store the number of terms in nterms.
- If the number of terms is more than 2, we use a while loop.

=> While loop in python :-

- While loop is used to execute a block of statements repeatedly until a given condition is satisfied.
- When condition becomes false, this line immediately after the loop is ~~exe~~ executed.

Syntax : While expression :
statement (s).

Program:

```
num = int(input("Enter the number: "))
a = 0
b = 1
for i in range(num):
  print(a)
  c = a + b
  a = b
  b = c
```

Output:

```
PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1e.py
Enter the number: 7
0
1
1
2
3
5
8
```

F) WAP to program the following pattern

Program:

```
# pattern drawing
terms = int(input("Enter the number of terms: "))

for i in range(terms):
  for j in range(i+1):
    print(chr(i+65), end="")
  print()

for i in range(terms,0,-1):
  for j in range(terms-i):
    print(" ", end="")
  for j in range(i):
    print("*", end="")
  print()

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

for i in range(terms):
  for j in range(terms-i):
    print(" ", end="")
  for j in range(i+1):
    print(" *", end="")
  print()
```

Output:


PS C:\Users\harsh\Desktop\College\Python lab> python exp_1/1f.py
Enter the number of terms: 7
A
BB
CCC
DDDD
EEEEE
FFFFFF
GGGGGGG
****
******
*****
****
***
**
*
1
121
12321
1234321
123454321
12345654321
1234567654321
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *