Problem no: 01

Factorial determination using Python Functions.

Program Title:

Calculating **factorial** of an integer using **user defined python function**.

Objectives:

To calculate factorial of an integer.

Theory:

In all programming and scripting language, a function is a block of program statements which can be used repetitively in a program.

* In Python, a user-defined function's declaration begins with the keyword def and followed by the function name.
* The function may take arguments(s) as input within the opening and closing parentheses, just after the function name followed by a colon.
* After defining the function name and arguments(s) a block of program statement(s) start at the next line and these statement(s) must be indented.

def function\_name(argument1, argument2, ...) :

    statement\_1

    statement\_2

    ....

Code:

1. **def** factorial(n):
2. **if** n == 0 **or** n == 1:
3. **return** 1
4. **else**:
5. **return** n \* factorial(n - 1)
7. **while** True:
8. n = int(input('Enter a number: '))
9. **if** n == 0:
10. **break**
11. **else**:
12. fact = factorial(n)
13. **print**('Factorial of ', n, ' is ', fact)

Output:

Enter a number: 10

Factorial of 10 is 3628800

Enter a number: 0

Problem no: 02

String Matching

Program Title:

Write a python program that matches a string that has an 'a' followed by anything ending 'b'.

Objectives:

To learn string matching using regex.

Theory:

**2.1 re.search ():**

The search() function searches the string for a match, and returns a Match object if there is a match.

If there is more than one match, only the first occurrence of the match will be returned.

Code:

1. **import** re
2. **import** numpy as np
4. **def** patern\_match(text):
5. patterns = 'a.\*b$'
6. **if** re.search(patterns, text):
7. **return** 1
8. **else**:
9. **return** 0
11. data = np.array(['ab', 'aabb', 'abcd', 'abababc', 'darb', 'bab', 'sa job'])
13. **for** i **in** data:
14. flag = patern\_match(i)
15. **if** flag == 1:
16. **print**(i, ' is matched')
17. **else**:
18. **print**(i, 'is not matched')

Output:

ab is matched

aabb is matched

abcd is not matched

abababc is not matched

darb is matched

bab is matched

sa job is matched

Program Title:

Write a python program that finds scores and wickets of two teams from a given text.

Theory:

**2.1 re.split ():**

The split() function returns a list where the string has been split at each match

Code:

1. data = '''''India played U-19 world cup final in 2006, 2008, 2010, 2020. Top scorers from BD U-19 team was Akbar Ali 43(59), Parvez 47(79), Rokibul 20(32). And from bowlers end Sakib 3-28, Shariful 2-51, Rokibul 2-37. And top players from IND U-19 team was Joysowal 88(121), Saxena 2(17), Veer 0(1). and from bowling end Tyagi 0-33, Mishra 2-25. they performed well too.'''
3. **import** re
5. Year\_pattern = re.compile(r'\d{4}')
6. years = Year\_pattern.findall(data)
7. **print**("IND played world cup finals:")
8. **print**(years)
10. **for** item **in** re.findall("\S.\*BD.\*\w\S", data):
11. year, stat = re.split("B", item)
13. **for** line **in** re.findall("\S.\*IND.\*\w\S", stat):
14. ban, ind = re.split("I", line)
16. Run\_pattern = re.compile(r'[0-9]+\([0-9]+\)')
17. Wicket\_pattern = re.compile(r'[0-9]\-[0-9]+')
19. **print**("\nBD stats:")
20. BD\_runs = Run\_pattern.findall(ban)
21. **print**(BD\_runs)
22. BD\_wickets = Wicket\_pattern.findall(ban)
23. **print**(BD\_wickets)
25. **print**("\nIND stats:")
26. IND\_runs = Run\_pattern.findall(ind)
27. **print**(IND\_runs)
28. IND\_wickets = Wicket\_pattern.findall(ind)
29. **print**(IND\_wickets)

Output:

IND played world cup finals:

['2006', '2008', '2010', '2020']

BD stats:

['43(59)', '47(79)', '20(32)']

['3-28', '2-51', '2-37']

IND stats:

['88(121)', '2(17)', '0(1)']

['0-33', '2-25']