Practical no. 1

Aim: Print only the words that start with letter ‘s’ in the following statement.

String Split(): Python String split() method in Python split a string into a list of strings after breaking the given string by the specified separator.

For loop: Python For loop is used for sequential traversal i.e. it is used for iterating over an iterable like String, Tuple, List,
Set or Dictionary.

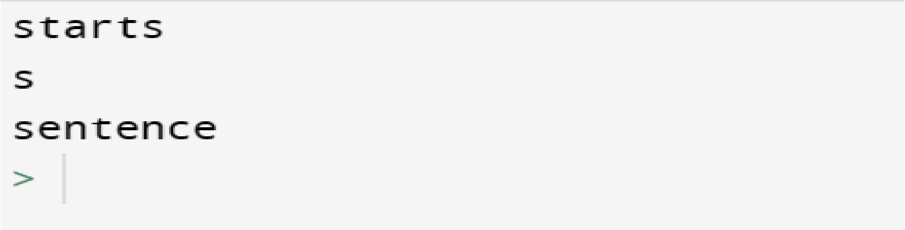
IF: The if statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not.

```
Syntax: if condition:
  statement1
statement2
```

Program:

```
s="print only the word that starts with s in this sentence"
for i in s.split():
    if i[0]=='s':
        print(i)
```

Output:



```
starts
s
sentence
>
```

Practical no. 2

Aim: Print every word from the below sentence which has even number of letters.

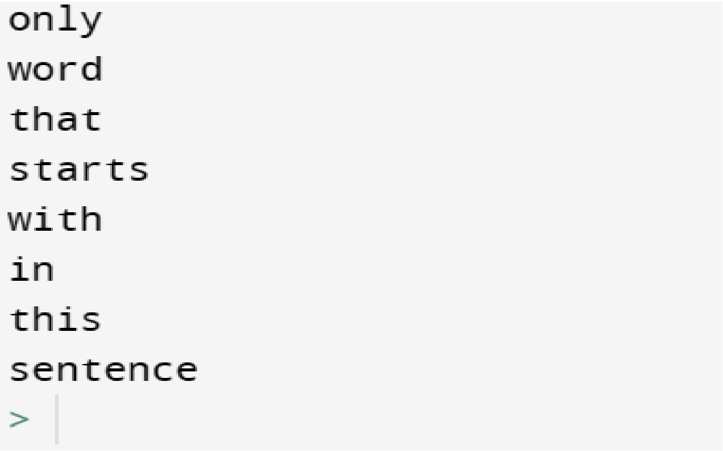
Len() function: Python len() function is an inbuilt function in Python. It can be used to find the length of an object.

Syntax: len(Object)

Program*:*

```
s="print only the word that starts with s in this sentence"
for i in s.split():
    if len(i) % 2 == 0:
        print(i)
```

Output:



```
only
word
that
starts
with
in
this
sentence
>
```

Practical no. 3

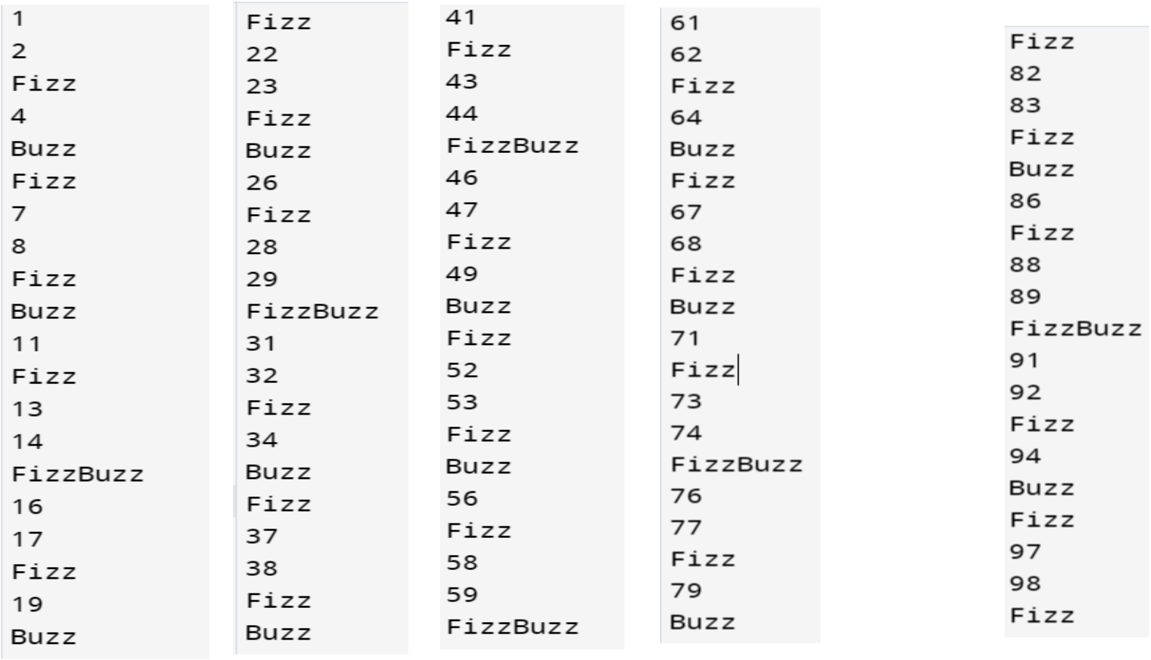
Aim: Write a program that prints the integers from 1 to 100, but for multiples of 3 print ‘FIZZ’ instead of number and for multiples of five print ‘BUZZ’. For numbers which are multiples of both 3 and 5 print ‘FIZZBUZZ’.

Elif function: The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

Program:

```
for fizzbuzz in range(1,100):
        if fizzbuzz % 15 == 0:
                print("FizzBuzz")
        continue
        elif fizzbuzz % 3 == 0:
                print("Fizz")
                continue
        elif fizzbuzz % 5 == 0:
                print("Buzz")
continue
                print(fizzbuzz)
```

Output:



```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
```

```
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
FizzBuzz
31
32
Fizz
34
Buzz
Fizz
37
38
Fizz
Buzz
```

```
41
Fizz
43
44
FizzBuzz
46
47
Fizz
49
Buzz
Fizz
52
53
Fizz
Buzz
56
Fizz
58
59
FizzBuzz
```

```
61
62
Fizz
64
Buzz
Fizz
67
68
Fizz
Buzz
71
Fizz
73
74
FizzBuzz
76
77
Fizz
79
Buzz
```

```
Fizz
82
83
Fizz
Buzz
86
Fizz
88
89
FizzBuzz
91
92
Fizz
94
Buzz
Fizz
97
98
Fizz
```

Practical no. 4

Aim: Write a program using function to check who is employee of the month.

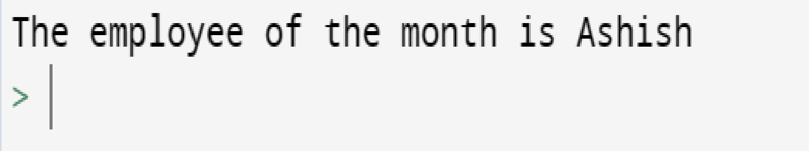
Def keyword: def keyword is used to define a function, it is placed before a function name that is provided by the user to create a user-defined function.

Syntax: def function_name(parameters):
            Fuction definition statements..

Program:

```
def employee_of_month(employees):
        best_employee = None
        best_score = 0
    for employee, score in employees.items():
      if score > best_score:
         best_employee = employee
         best_score = score
    return best_employee
    employees = {
                    'Ashish': 90,
                    'Prem': 80,
                    'Sahil': 85,
                    }
    print(f'The employee of the month is {employee_of_month(employees)}')
```

Output:



```
The employee of the month is Ashish
>
```

Practical no 5

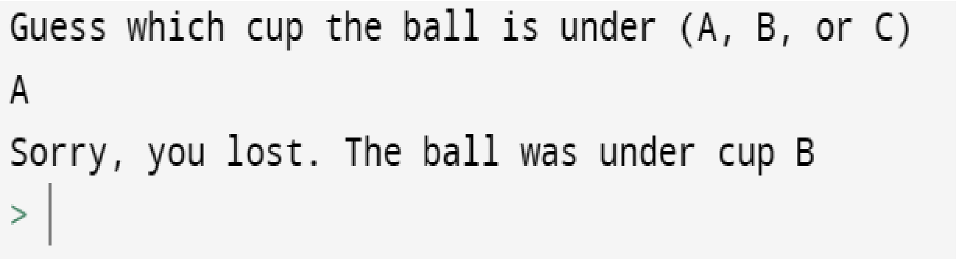
Aim: Write a program to mimic the carnival game ‘Three Cup Montee’.

Shuffle() function: The shuffle() function is used to shuffle a sequence (list).

Code:

```
import random
    def shuffle_cups(cups):
        random.shuffle(cups)
    def play_game():
        cups = ['A', 'B', 'C']
    shuffle_cups(cups)
    print('Welcome to Three Cup Montee!')
    print('Guess which cup the ball is under (A, B, or C)')
    guess = input().upper()
    if cups.index(guess) == 1:
      print('You won! The ball was under cup', guess)
   else:
       print('Sorry, you lost. The ball was under cup', cups[1])
   play_game()
```

Output:



```
Guess which cup the ball is under (A, B, or C)
A
Sorry, you lost. The ball was under cup B
> 
```

Practical no. 6

Aim: Write a program that returns the lesser of two given numbers if both numbers are even, but returns the greater if one or both numbers are odd.

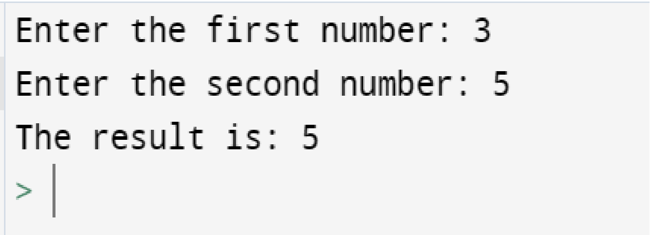
If-Else statement: The if statement tells us that if a condition is true it will execute a block . But if we want to do something else if the condition is false, we can use the else statement with if statement to execute a block of code when the if condition is false.

```
Syntax: if (condition):
            # Executes this block if condition is true
            else:
            # Executes this block if condition is false
```

Code:

```
def compare_numbers():
    a = int(input("Enter the first number: "))
    b = int(input("Enter the second number: "))
    if a % 2 == 0 and b % 2 == 0:
        return min(a, b)
    else:
        return max(a, b)
result = compare_numbers()
print("The result is:", result)
```

Output:



```
Enter the first number: 3
Enter the second number: 5
The result is: 5
> 
```

Practical no. 7

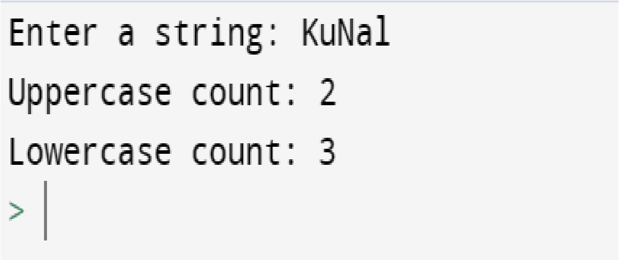
Aim: Write a python function that accepts a string and calculate the number of Upper case letters and lower case letters.

String: A string is a data structure in Python that represents a sequence of characters.

Code:

```
def count_upper_lower():
    string = input("Enter a string: ")
    upper_count = 0
    lower_count = 0
    for char in string:
        if char.isupper():
            upper_count += 1
        elif char.islower():
            lower_count += 1
    return (upper_count, lower_count)
result = count_upper_lower()
print("Uppercase count:", result[0])
print("Lowercase count:", result[1])
```

Output:



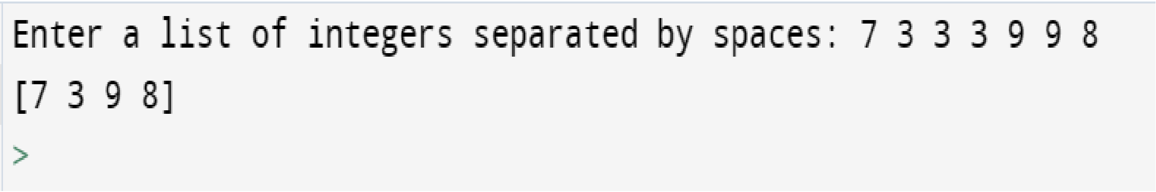
```
Enter a string: KuNal
Uppercase count: 2
Lowercase count: 3
> |
```

Practical no. 8

Aim: Write a python function that takes a list and return a new list with unique elements of the first list. For example,Sample List =[1,1,1,2,2,3,3,4]Unique List = [1,2,3,4].

Code:
```
def unique_list():
    lst = input("Enter a list of integers separated by spaces: ").split()
    lst = [int(item) for item in lst]
    unique_lst = []
    for item in lst:
        if item not in unique_lst:
            unique_lst.append(item)
    return unique_lst
result = unique_list()
print(result)
```

Output:



```
Enter a list of integers separated by spaces: 7 3 3 3 9 9 8
[7 3 9 8]
>
```

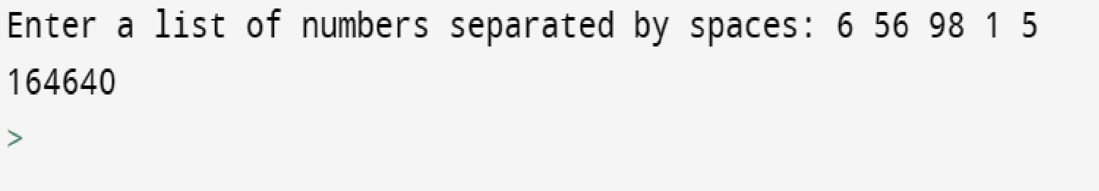
Practical no.9

Aim: Write a python function to multiply all the numbers in the list.

Code:

```
def multiply_list(liist):
    product = 1
    for num in liist:
        product *= num
    return product
liist = input("Enter a list of numbers separated by spaces: ").split()
liist = [int(item) for item in liist]
result = multiply_list(liist)
print(result)
```

Output:



```
Enter a list of numbers separated by spaces: 6 56 98 1 5
164640
>
```

Practical no.10

Aim: Write a program for validating the user input.

While loop: While Loop is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.

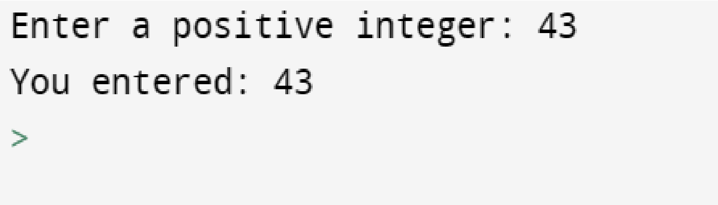
Syntax: while expression:

```
    statement(s)
```

Code:

```
while True:
    try:
        user_input = int(input("Enter a positive integer: "))
        if user_input <= 0:
            print("Please enter a positive integer.")
        else:
            break
    except ValueError:
        print("Invalid input. Please enter a positive integer.")
print("You entered:", user_input)
```

Output:



```
Enter a positive integer: 43
You entered: 43
>
```

Practical no.11

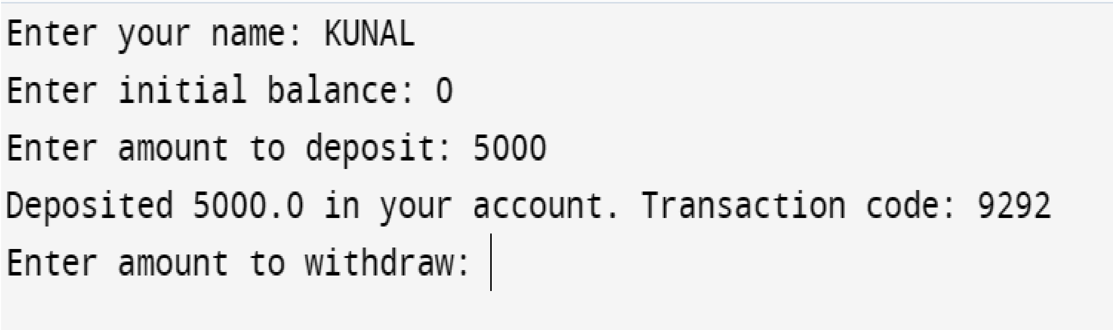
Aim: Using Object oriented Programming, write a program for opening a Bank account, deposit of money and withdrawal of money. Also generate a 4 digit unique code for each transaction.

Random function(): Functions in the random module rely on a pseudo-random number generator function random(), which generates a random number.

Code:

```
import random
class BankAccount:
    def __init__(self, name, balance):
        self.name = name
        self.balance = balance
        self.account_number = random.randint(1000, 9999)
    def deposit(self, amount):
        self.balance += amount
        transaction_code = random.randint(1000, 9999)
        print(f"Deposited {amount} in your account. Transaction code: {transaction_code}")
        return transaction_code
    def withdraw(self, amount):
        if self.balance < amount:
            print("Insufficient balance")
            return None
        self.balance -= amount
        transaction_code = random.randint(1000, 9999)
        print(f"Withdrawn {amount} from your account. Transaction code: {transaction_code}")
        return transaction_code
    def check_balance(self):
        print(f"Your account balance is {self.balance}")
name = input("Enter your name: ")
balance = float(input("Enter initial balance: "))
account = BankAccount(name, balance)
deposit_amount = float(input("Enter amount to deposit: "))
deposit_code = account.deposit(deposit_amount)
withdraw_amount = float(input("Enter amount to withdraw: "))
withdraw_code = account.withdraw(withdraw_amount)
account.check_balance()
```

Output:



```
Enter your name: KUNAL
Enter initial balance: 0
Enter amount to deposit: 5000
Deposited 5000.0 in your account. Transaction code: 9292
Enter amount to withdraw: 
```

Practical no.12

Aim: Write a program to print next 5 days starting from today.

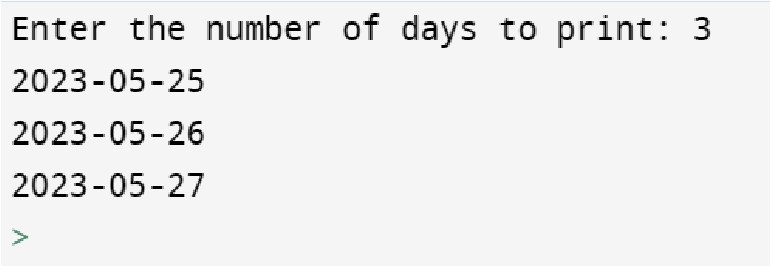
Datetime module: Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

Constructor Syntax: class datetime.date(year, month, day)

Code:

```
import datetime
today = datetime.date.today()
num_days = int(input("Enter the number of days to print: "))
for i in range(num_days):
    next_day = today + datetime.timedelta(days=i+1)
    print(next_day)
```

Output:



```
Enter the number of days to print: 3
2023-05-25
2023-05-26
2023-05-27
>
```

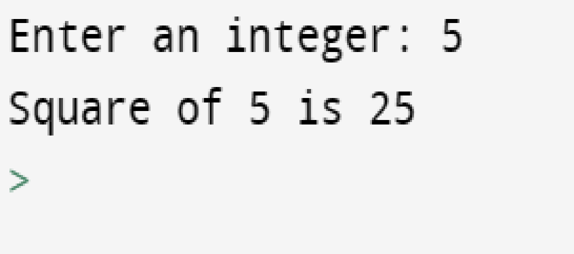
Practical no.13

Aim: Write a function that asks for an integer and prints square of it. Use a while loop with a try, except, else block to account for incorrect inputs.

Code:

```
while True:
    try:
        num = int(input("Enter an integer: "))
    except ValueError:
        print("Invalid input! Please enter an integer.")
    else:
        print("Square of", num, "is", num*num)
        break
```

Output:



```
Enter an integer: 5
Square of 5 is 25
>
```