For Loop Example:

temp.py

```
1   m = [1, 2, 3, 4, 5]
2   n = [3, 4, 5, 6, 7]
3   z = []
4
5   for i in range(len(m)):
6       for j in range (len(n)):
7           if(m[i]==n[j]):
8               z.append(m[i])
9
10
11  print(z)
12
```

Output:

```
In [15]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')
[3, 4, 5]
```

Simple Calculator using if Condition:

temp.py

```
1   print("Please enter the two values to perform calculation")
2   a=int(input())
3   b=int(input())
4   print("please enter 'a' for addition 's' for subtraction 'm' for multiplication 'd' for Division")
5   p=input()
6
7   if p in ['a', 'A']:
8       print(a+b)
9   elif p in ['s', 'S']:
10      print(a-b)
11  elif p in ['m', 'M']:
12      print(a*b)
13  elif p in ['d', 'D']:
14      print(a/b)
15  else:
16      print("Enter appropriate input for further calculation")
17
```

Output:

```
In [22]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')
Please enter the two values to perform
calculation

3

55
please enter 'a' for addition 's' for
subtraction 'm' for multiplication 'd'
for Division

D
0.05454545454545454
```

Web scrapping code:

temp.py

```python
import requests

response = requests.get('https://en.wikipedia.org/wiki/List_of_rivers_of_India')

result = {
    'status_code': response.status_code,
    'headers': dict(response.headers),
    'content': response.text
}

print(result)

```

Using Dictionary and list with For and If loop:

temp.py

```python
people = [
    {'name': 'Ajay', 'age': 25},
    {'name': 'Ragav', 'age': 30},
    {'name': 'Charlie', 'age': 35},
    {'name': 'Raj', 'age': 40},
    {'name': 'Smith', 'age': 45}
]

names_over_30 = []

for person in people:
    if person['age'] > 30:
        names_over_30.append(person['name'])

print(names_over_30)

```

Output:

```
In [11]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')
['Charlie', 'Raj', 'Smith']
```

Resolving error of printing loop of percentage :

```
max_marks = [100, 75, 100, 35, 50]
marks = {
   "stu1": [55, 35, 65, 20, 33],
   "stu2": [95, 35, 70, 25, 36],
   "stu3": [50, 60, 65, 22, 45],
   "stu4": [58, 65, 87, 16, 54],
   "stu5": [35, 44, 76, 24, 49],
   "stu6": [22, 54, 54, 10, 35],
   "stu7": [56, 56, 86, 6, 45],
   "stu8": [75, 74, 88, 5, 5],
   "stu9": [99, 32, 24, 4, 24],
   "stu10": [13, 45, 86, 33, 34],
   "stu11": [56, 54, 65, 23, 43],
   "stu12": [35, 67, 56, 32, 40],
   "stu13": [45, 13, 36, 30, 44],
   "stu14": [29, 45, 22, 23, 33],
   "stu15": [59, 24, 65, 34, 23]
}

for name, marks_list in marks.items():
   per_stu = []
   print(name, "marks:", marks_list)
   for i, j in zip(marks_list, max_marks):
      per = round((i / j) * 100)
      per_stu.append(per)
      if per < 100 and per > 80:
         print("Name : " + name + ", Percentage : " + str(per))
         print("Grade : A")
      elif per <= 80 and per > 60:
         print("Name : " + name + ", Percentage : " + str(per))
         print("Grade : B")
      elif per <= 60 and per > 50:
         print("Name : " + name + ", Percentage : " + str(per))
         print("Grade : C")
      elif per <= 50:
         print("Name : " + name + ", Percentage : " + str(per))
         print("Grade : D")
   avg_per = sum(per_stu) / len(per_stu)
   print("Average Percentage for " + name + ":", avg_per)
```

Output:

stu1 marks: [55, 35, 65, 20, 33]
percentage of student 55
Name : stu1, Percentage : 55
Grade : C
percentage of student 47
Name : stu1, Percentage : 47
Grade : D
percentage of student 65
Name : stu1, Percentage : 65
Grade : B
percentage of student 57
Name : stu1, Percentage : 57
Grade : C
percentage of student 66
Name : stu1, Percentage : 66
Grade : B
Average Percentage for stu1: 58.0

Logical Programs:

1.Palindrome :

temp.py*

```
1  s = input("Enter a string: ")
2  if (s==s[::-1]):
3      print("The string is a palindrome.")
4  else:
5      print("The string is not a palindrome.")
6
```

Output:

```
In [7]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')

Enter a string: baba
The string is not a palindrome.

In [8]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')

Enter a string: jaaj
The string is a palindrome.
```

2.Prime number check:

temp.py

```
import math
def Prime_Num():
    if a<2:
        return False
    elif a>2:
        for i in range(2,math.ceil(math.sqrt(a))):
            if a%i==0:
                return False
            else:
                return True
    else:
        return True

a=int(input("Enter a number:"))
if Prime_Num():
    print("The num is prime")
else:
    print("It is not a prime")
```

Output:

```
In [15]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')

Enter a number:2
The num is prime

In [16]: runfile('C:/Users/admin/.spyder-
py3/temp.py', wdir='C:/Users/
admin/.spyder-py3')

Enter a number:6
It is not a prime
```

SQL Practice:

- **Creating database:**

```
1 •  create database practice;
```

- **Creating tables:**

```
2 •  use practice;
3 •  create table student(Student_id int,Student_Name char,Student_Class int);
4 •  alter table student
5    Modify Student_Name Text;
6 •  insert into  student(Student_id ,Student_Name ,Student_Class) values(1,"Raj",3);
7 •  insert into  student(Student_id ,Student_Name ,Student_Class) values(2,"Mahesh",3);
8 •  insert into  student(Student_id ,Student_Name ,Student_Class) values(3,"Ramesh",3);
9 •  insert into  student(Student_id ,Student_Name ,Student_Class) values(1,"Sravan",4);
10 • insert into  student(Student_id ,Student_Name ,Student_Class) values(2,"roshan",4);
```

- **Operations:**

```
11 • select * from student where Student_id=1;
```

Student_id	Student_Name	Student_Class
1	Raj	3
1	Sravan	4

```
12 • select Student_id from student where Student_Name Like "R%";
```

Student_id
1
3
2

```
13    Update student
14    set Student_Name="Lokesh" Where Student_id=1 and Student_Class=1;
```

```
15    select Min(Student_id) from student
16    where Student_Class=3;
```

	Min(Student_id)
▶	1

1