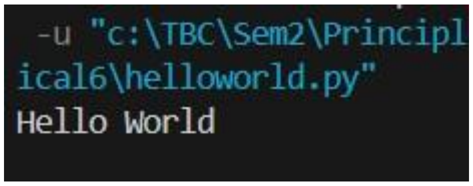
1. Task1 print("Hello World \n") Output:

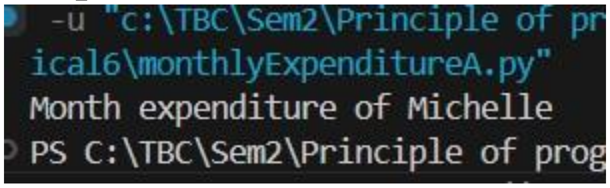


```
 -u "c:\TBC\Sem2\Principl
ical6\helloworld.py"
Hello World
```

2. Monthly Expenditure A:

```
#Practical 6 part 2 print ("Month expenditure of
{}".format("Michelle"))
```

Output:

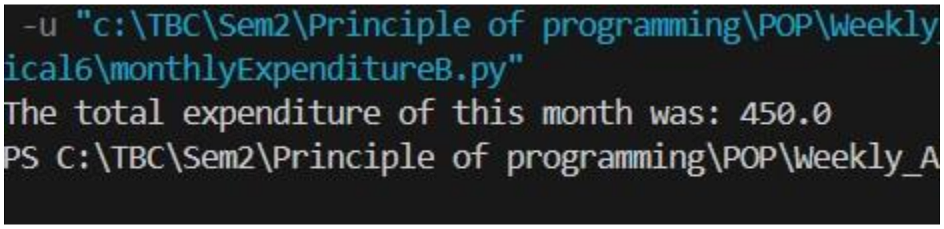


```
  -u "c:\TBC\Sem2\Principle of pr
ical6\monthlyExpenditureA.py"
Month expenditure of Michelle
PS C:\TBC\Sem2\Principle of prog
```

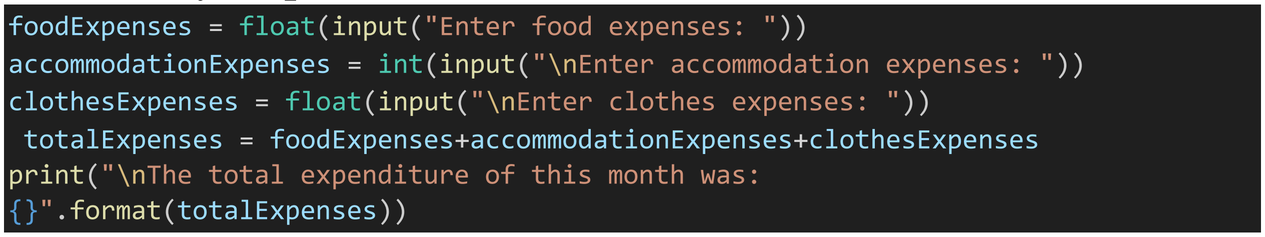
3. Monthly Expenditure B

```
foodExpenses = 300.0
leisureExpenses = 100.0 #assign 100.0 to leisureExpenses
clothesExpenses = 50.0 #assign 50.0 to clothesExpenses
totalSpent = 0.0 # float variable for total expenses, initialised to 0.

totalSpent = foodExpenses + leisureExpenses + clothesExpenses
print("The total expenditure of this month was: {}".format(totalSpent))
```

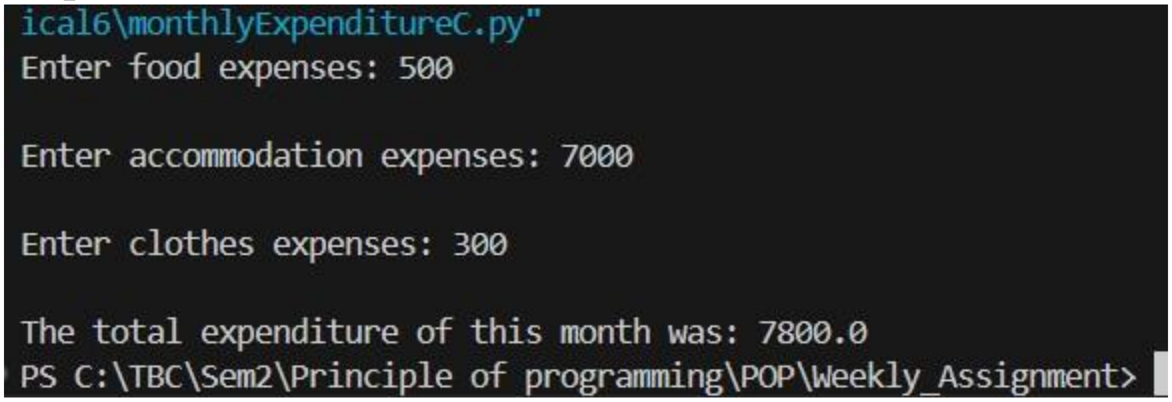
Output:


```
 -u "c:\TBC\Sem2\Principle of programming\POP\Weekly
ical6\monthlyExpenditureB.py"
The total expenditure of this month was: 450.0
PS C:\TBC\Sem2\Principle of programming\POP\Weekly_A
```

4. Monthly Expenditure C:


```python
foodExpenses = float(input("Enter food expenses: "))
accommodationExpenses = int(input("\nEnter accommodation expenses: "))
clothesExpenses = float(input("\nEnter clothes expenses: "))
 totalExpenses = foodExpenses+accommodationExpenses+clothesExpenses
print("\nThe total expenditure of this month was:
{}".format(totalExpenses))
```

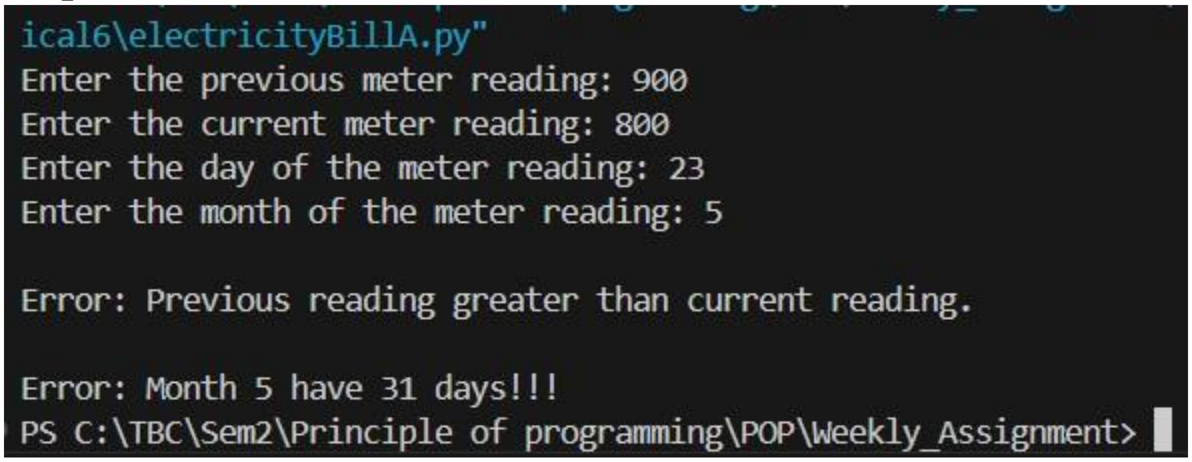
Output:


```
ical6\monthlyExpenditureC.py"
Enter food expenses: 500

Enter accommodation expenses: 7000

Enter clothes expenses: 300

The total expenditure of this month was: 7800.0
PS C:\TBC\Sem2\Principle of programming\POP\Weekly_Assignment>
```

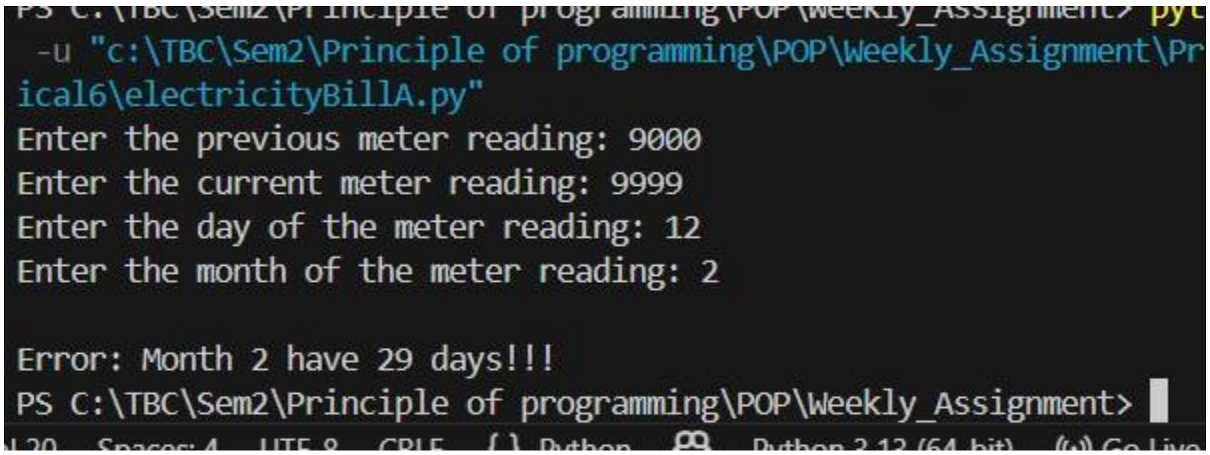
5. Validating Electricity Reading

```
previousMR = int(input("Enter the previous meter reading: "))
previousMR = int(input("Enter the previous meter reading: "))
currentMR = int(input("Enter the current meter reading: "))
day = int(input("Enter the day of the meter reading: ")) month
= int(input("Enter the month of the meter reading: "))
 if(previousMR<0 or
previousMR>9999):
    print("\nError: Previous meter reading out of range!!!\n")
 if(currentMR<0 or
currentMR>9999):
    print("\nError: Curremt meter reading out of range!!!\n")
 if(previousMR>currentMR):
    print("\nError: Previous reading greater than current reading.\n")
else:
    electricity_used  =   currentMR-previousMR
if (electricity_used>1000):
        print("\nError: Electricity used is more than 1000\n")
     if month<1 or
month >12:
    print("Error: Month should be 1-12!!")
else:
    #Months which have 31 days
if month in [1,3,5,6,8,10,12]:
        #if days is not 31
if day!=31:
            print(f"\nError: Month {month} have 31 days!!!")
    #Months which have 30
days     if month in
[4,6,9,11]:          #if days
is not 30         if day!=30:
            print(f"\nError: Month {month} have 30 days!!!")
    #Months which have 29 days
if month ==2:
        #if days is not 29
if day!=29:
            print(f"\nError: Month {month} have 29 days!!!")
```

Output:



```
ical6\electricityBillA.py"
Enter the previous meter reading: 900
Enter the current meter reading: 800
Enter the day of the meter reading: 23
Enter the month of the meter reading: 5

Error: Previous reading greater than current reading.

Error: Month 5 have 31 days!!!
PS C:\TBC\Sem2\Principle of programming\POP\Weekly_Assignment>
```

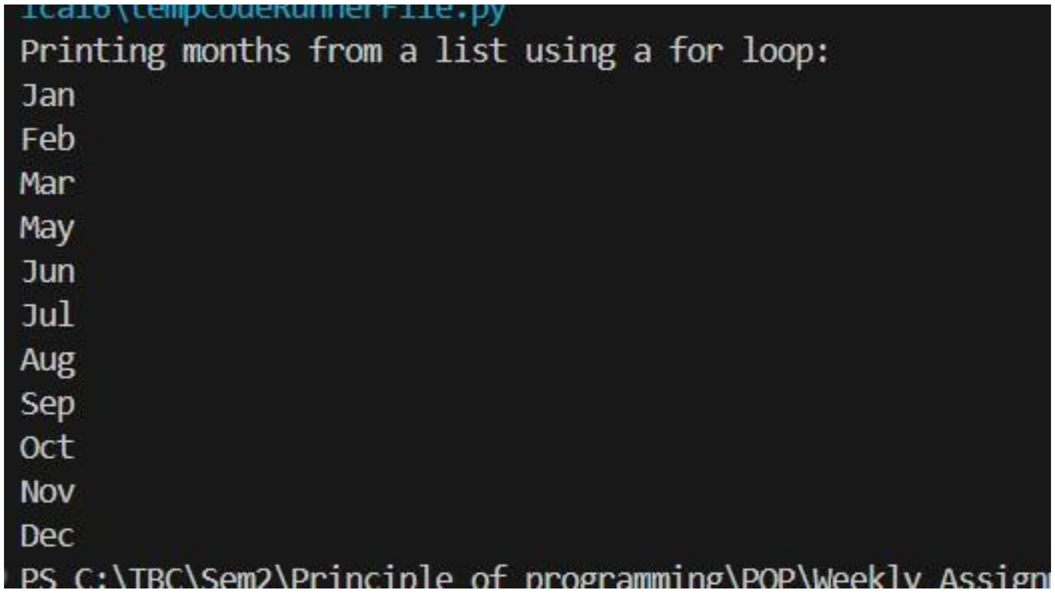


```
PS C:\TBC\Sem2\Principle of programming\POP\Weekly_Assignment> pyt
 -u "c:\TBC\Sem2\Principle of programming\POP\Weekly_Assignment\Pr
ical6\electricityBillA.py"
Enter the previous meter reading: 9000
Enter the current meter reading: 9999
Enter the day of the meter reading: 12
Enter the month of the meter reading: 2

Error: Month 2 have 29 days!!!
PS C:\TBC\Sem2\Principle of programming\POP\Weekly_Assignment>
```

6. Printing months from a list using for loop

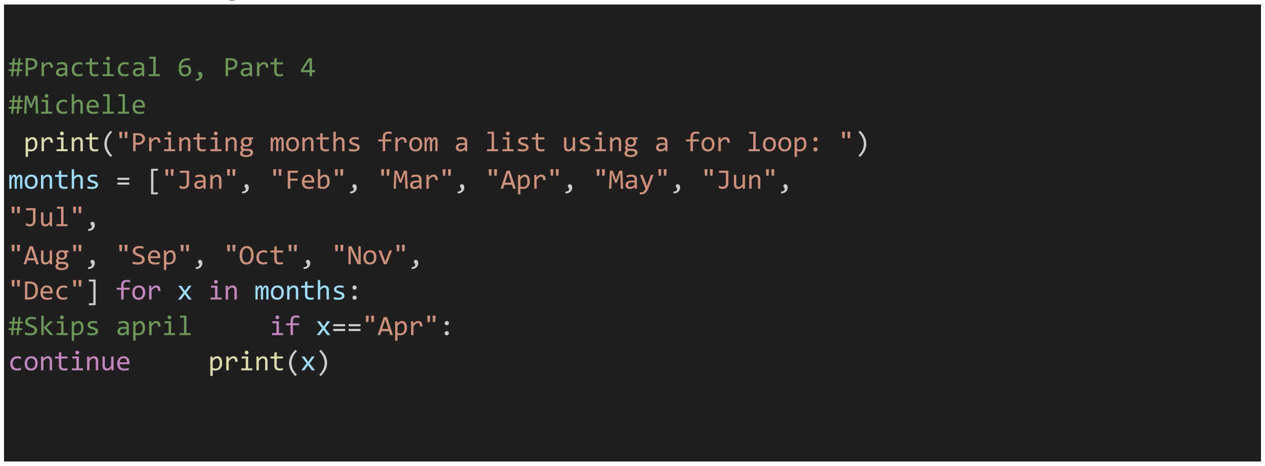
```
#Practical 6, Part 4
#Michelle
 print("Printing months from a list using a for loop: ")
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul",
"Aug", "Sep", "Oct", "Nov",
"Dec"] for x in months:
#Skips april     if x=="Apr":
continue     print(x)
```

Output:



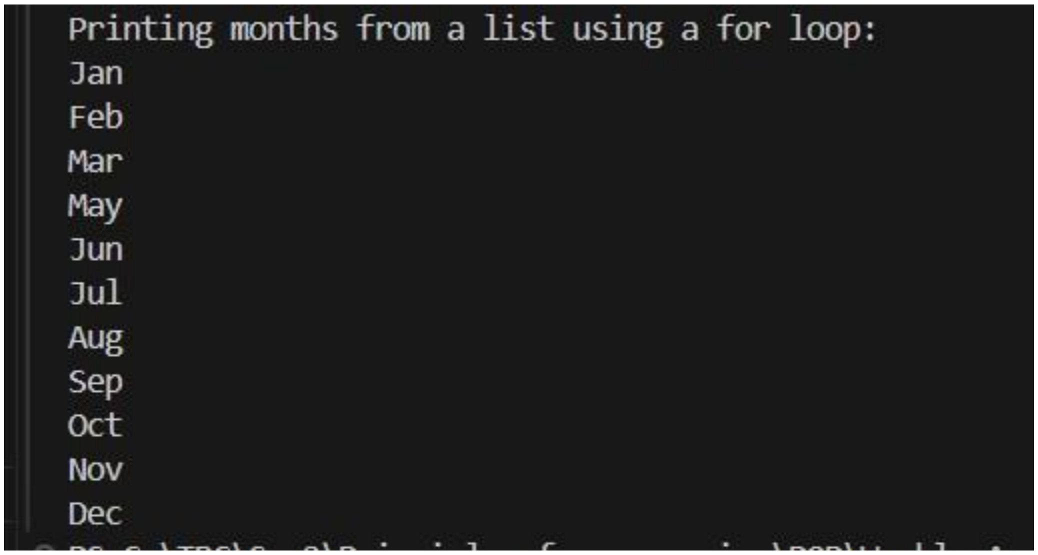
```
Printing months from a list using a for loop:
Jan
Feb
Mar
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
PS C:\TBC\Sem2\Principle of programming\POP\Weekly Assign
```

7. Printing Months



```
#Practical 6, Part 4
#Michelle
 print("Printing months from a list using a for loop: ")
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul",
"Aug", "Sep", "Oct", "Nov",
"Dec"] for x in months:
#Skips april     if x=="Apr":
continue     print(x)
```

Output:

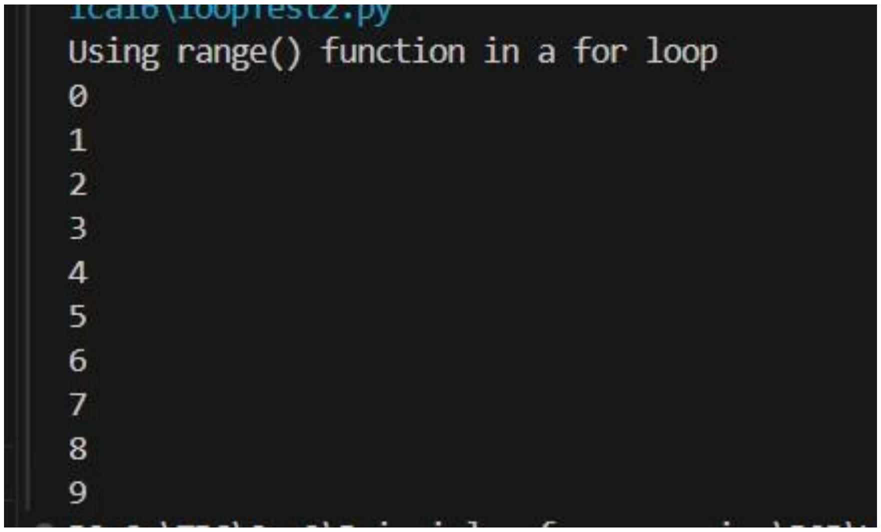


```
Printing months from a list using a for loop:
Jan
Feb
Mar
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
```

8. Priting number: Exercise 3

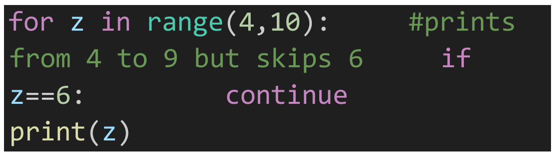
```
#Michelle print("Using range() function in a
for loop") for x in range(10):
    #by default start from 1, and increases by 1 step
print(x)
```

Output:



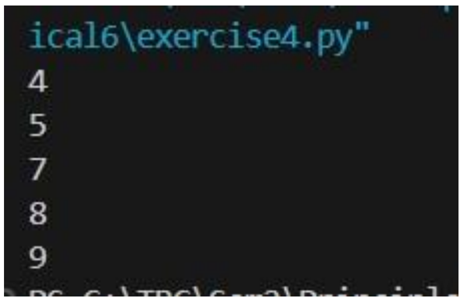
```
ica18\loopTest2.py
Using range() function in a for loop
0
1
2
3
4
5
6
7
8
9
```

9. Printing numbers



```
for z in range(4,10):     #prints
from 4 to 9 but skips 6     if
z==6:          continue
print(z)
```

Output:

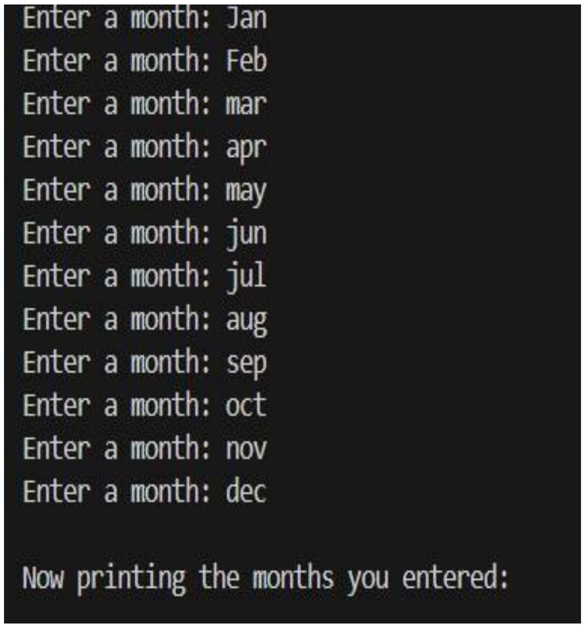


```
ical6\exercise4.py"
4
5
7
8
9
```

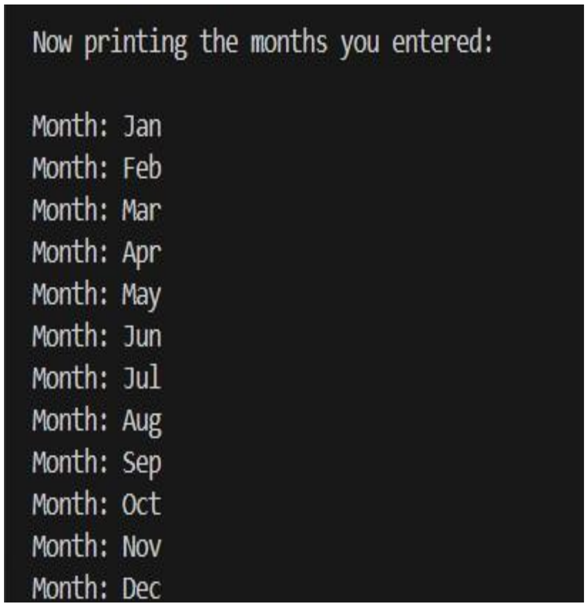
10. Taking input and printing

```
"""months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec"]
    for i in range (12):
    print(f"Month: {months[i]}")"""

months = []
index = 0

   #While loop for taking input
   while index<12:
   m = input("Enter a month: ").capitalize()
   months.append(m)
        index+=1

    #For loop for printing
    print("\nNow printing the months you entered: \n")
    for x in range(12):
    print(f"Month: {months[x]}")
```

Output:

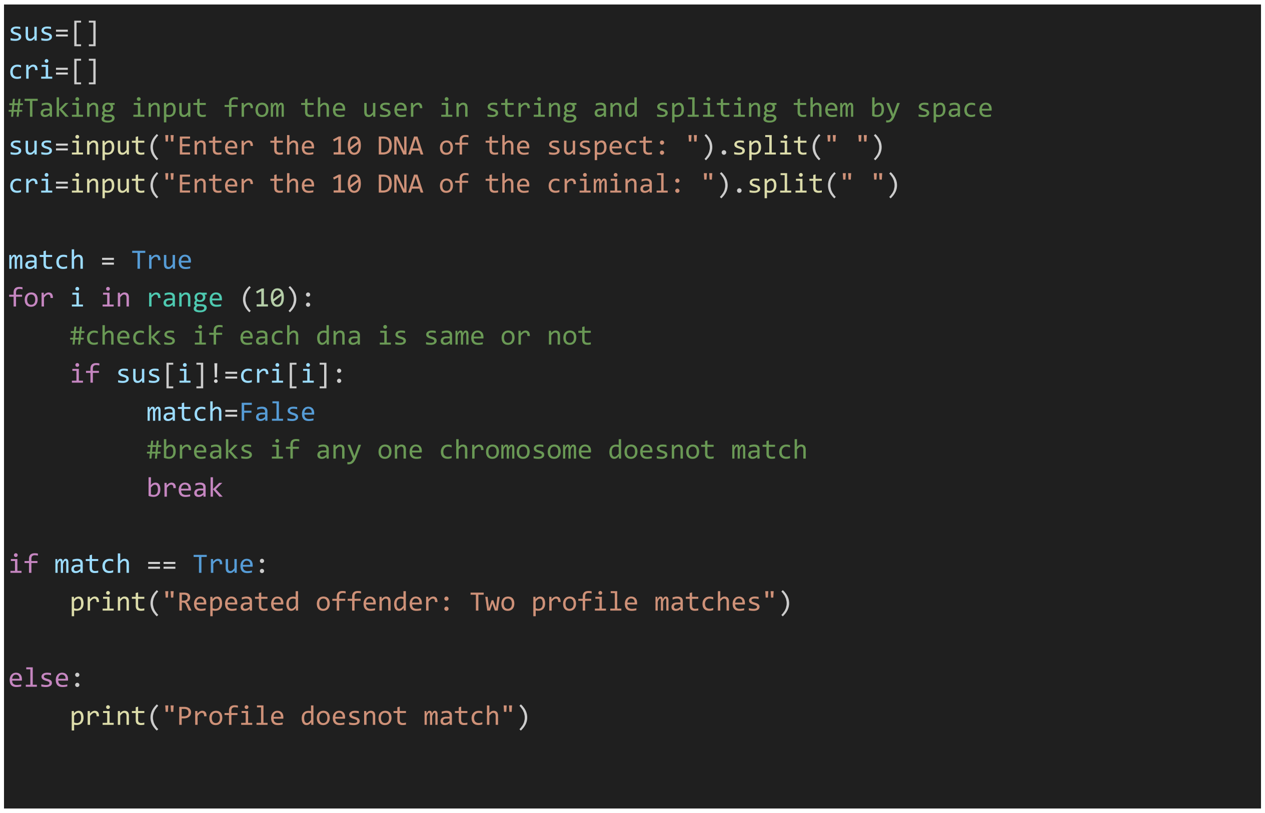


```
Enter a month: Jan
Enter a month: Feb
Enter a month: mar
Enter a month: apr
Enter a month: may
Enter a month: jun
Enter a month: jul
Enter a month: aug
Enter a month: sep
Enter a month: oct
Enter a month: nov
Enter a month: dec

Now printing the months you entered:
```



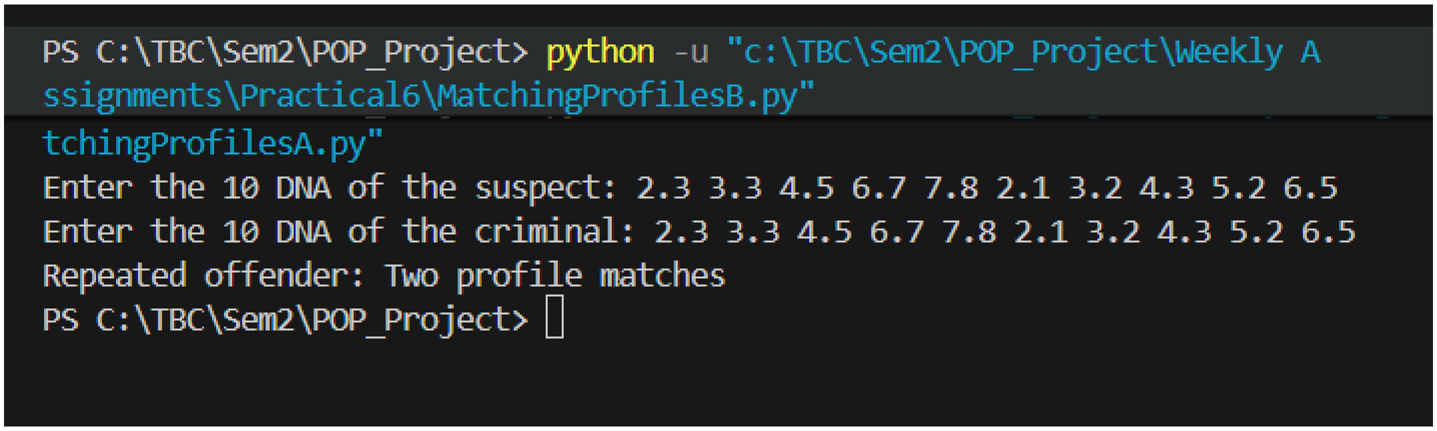
```
Now printing the months you entered:

Month: Jan
Month: Feb
Month: Mar
Month: Apr
Month: May
Month: Jun
Month: Jul
Month: Aug
Month: Sep
Month: Oct
Month: Nov
Month: Dec
```

11. Repeated offender:



```python
sus=[]
cri=[]
#Taking input from the user in string and spliting them by space
sus=input("Enter the 10 DNA of the suspect: ").split(" ")
cri=input("Enter the 10 DNA of the criminal: ").split(" ")

match = True
for i in range (10):
    #checks if each dna is same or not
    if sus[i]!=cri[i]:
         match=False
         #breaks if any one chromosome doesnot match
         break

if match == True:
    print("Repeated offender: Two profile matches")

else:
    print("Profile doesnot match")
```

Output:

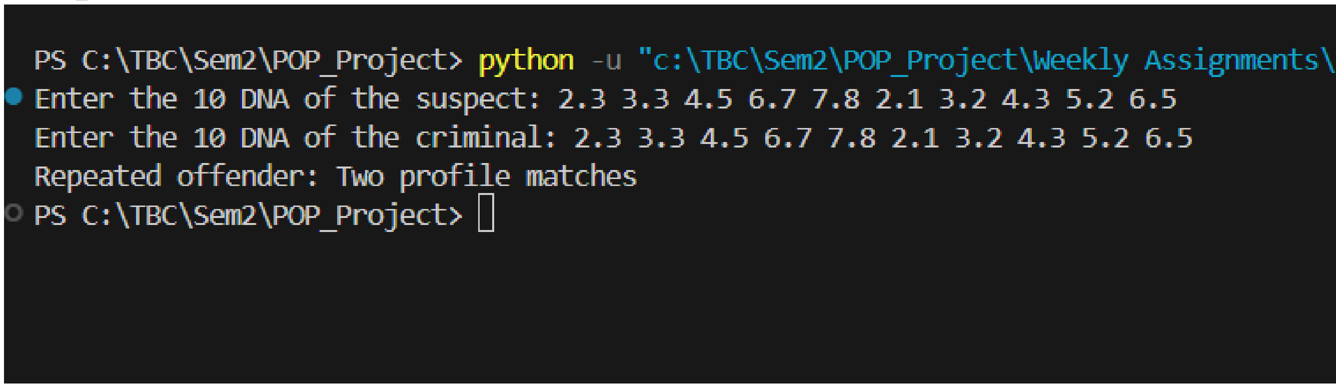


```
PS C:\TBC\Sem2\POP_Project> python -u "c:\TBC\Sem2\POP_Project\Weekly A
ssignments\Practical6\MatchingProfilesB.py"
tchingProfilesA.py"
Enter the 10 DNA of the suspect: 2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 DNA of the criminal: 2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Repeated offender: Two profile matches
PS C:\TBC\Sem2\POP_Project> 
```

12. Repeated offender using function

```
 #Matching profile B

 def user_input():

    sus=[]
    cri=[]
    #Taking input from the user in string and spliting them by space
    sus=input("Enter the 10 DNA of the suspect: ").split(" ")
    cri=input("Enter the 10 DNA of the criminal: ").split(" ")
    return sus,cri

def matchingProfiles(cri,sus):
    match = True
    for i in range (10):
        #checks if each dna is same or not
        if sus[i]!=cri[i]:
            match=False
            #breaks if any one chromosome doesnot match
            break
    return match

def main():
    sus_Dna, cri_Dna = user_input()
    match= matchingProfiles(sus_Dna,cri_Dna)
    if match == True:
        print("Repeated offender: Two profile matches")
    else:
        print("Profile doesnot match")

main()
```

Output:



```
PS C:\TBC\Sem2\POP_Project> python -u "c:\TBC\Sem2\POP_Project\Weekly Assignments\
Enter the 10 DNA of the suspect: 2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 DNA of the criminal: 2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Repeated offender: Two profile matches
PS C:\TBC\Sem2\POP_Project>
```