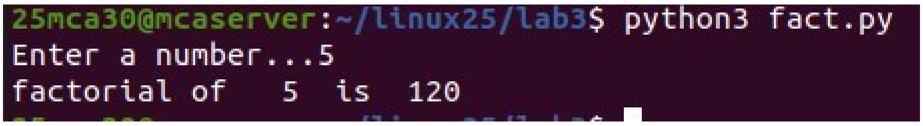
LAB CYCLE 3(1-10)

Name :Kiran Benny

Roll No:31

1. Write a program to find the factorial of a number

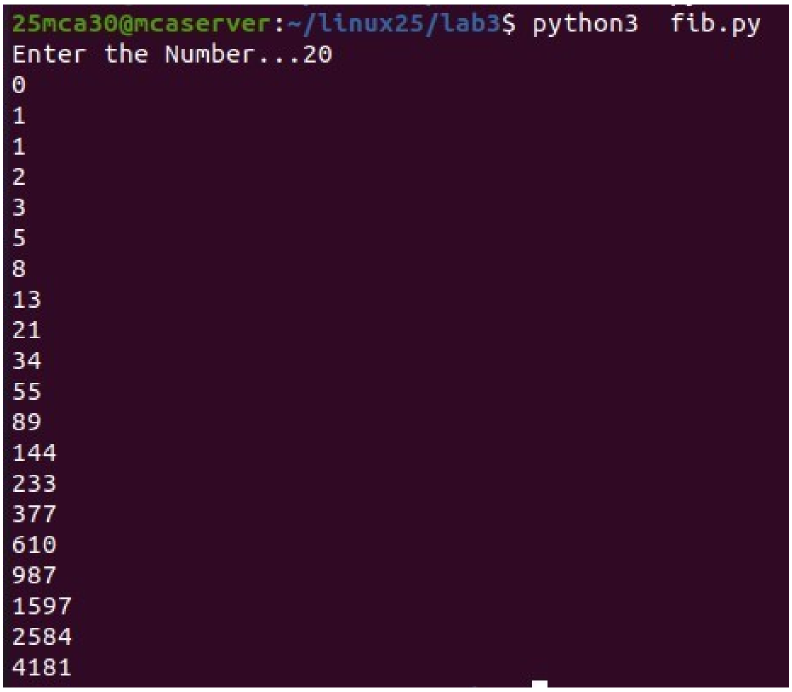
```
n=int(input("Enter a number..."))
def fact(n):
        if n==1:
                return 1
        else:
                return n*fact(n-1)

print("factorial of  ",n," is ",fact(n))
```



```
25mca30@mcaserver:~/linux25/lab3$ python3 fact.py
Enter a number...5
factorial of   5  is  120
```

2. Generate Fibonacci series of N terms

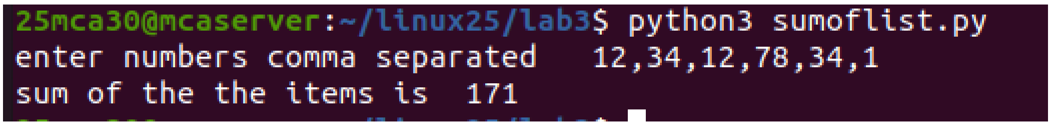
```
n=int(input("Enter the Number..."))
a=0
b=1
print(a)
print(b)
for i in range(2,n):
        c=a+b;
        print(c)
        a=b
        b=c
```



```
25mca30@mcaserver:~/linux25/lab3$ python3  fib.py
Enter the Number...20
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
```

3. Write a program to find the sum of all items in a list. [Using for loop]

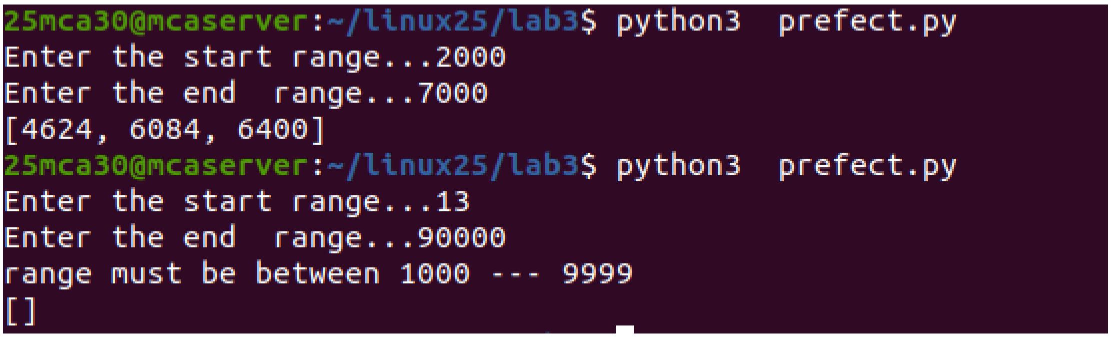
```
n=input("enter numbers comma separated  ")
numbers1=list(map(int,n.split(",")))
sum=0
for i in numbers1:
        sum=sum+i;
print("sum of the the items is ",sum)
```



```
25mca30@mcaserver:~/linux25/lab3$ python3 sumoflist.py
enter numbers comma separated   12,34,12,78,34,1
sum of the the items is  171
```

4. Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

```
import math
def even(n):
    for digits in str(n):
        if int(digits)%2!=0:
            return False
    return True

perfect_square=[]
sran=int(input("Enter the start range..."))
ran=int(input("Enter the end  range..."))
if ran>9999 or sran<1000:
    print("range must be between 1000 --- 9999")
else:
    for i in range(sran,ran):
        square=math.isqrt(i)
        if square*square==i:
            if even(i):
                perfect_square.append(i)

print(perfect_square)
```

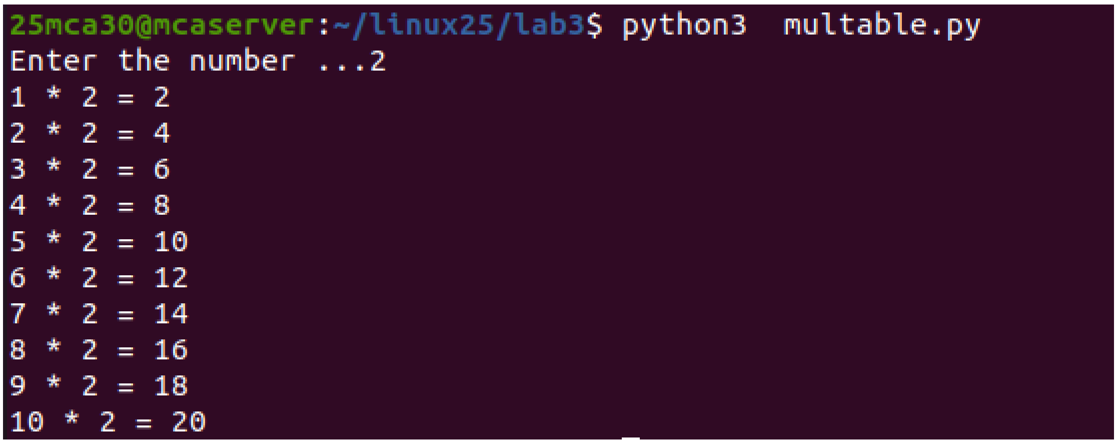


```
25mca30@mcaserver:~/linux25/lab3$ python3  prefect.py
Enter the start range...2000
Enter the end  range...7000
[4624, 6084, 6400]
25mca30@mcaserver:~/linux25/lab3$ python3  prefect.py
Enter the start range...13
Enter the end  range...90000
range must be between 1000 --- 9999
[]
```

5. Write a program using a for loop to print the multiplication table of n, where n is entered by the user.

```
n=int(input("Enter the number ..."))
```

```
for i in range(1,11):
        print(i ,"*",n,"=",i*n
        )
```

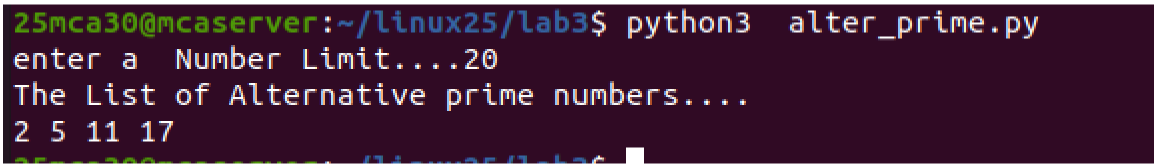


```
25mca30@mcaserver:~/linux25/lab3$ python3  multable.py
Enter the number ...2
1 * 2 = 2
2 * 2 = 4
3 * 2 = 6
4 * 2 = 8
5 * 2 = 10
6 * 2 = 12
7 * 2 = 14
8 * 2 = 16
9 * 2 = 18
10 * 2 = 20
```

6. Write a program to display alternate prime numbers till N (obtain N from the

```
user). def is_prime(num):
        if num<=1:
                return False
        for i in range(2,int(num ** 0.5)+1):
                if num%i==0:
                        return False
        return True



def alt_prime(num):
        count=0
        for i in range(2,num+1):
                if is_prime(i):
                        count+=1
                        if count%2!=0:
                                print(i,end=" ")



n=int(input("enter a  Number Limit...."))
print("The List of Alternative prime numbers…")
alt_prime(n)
```

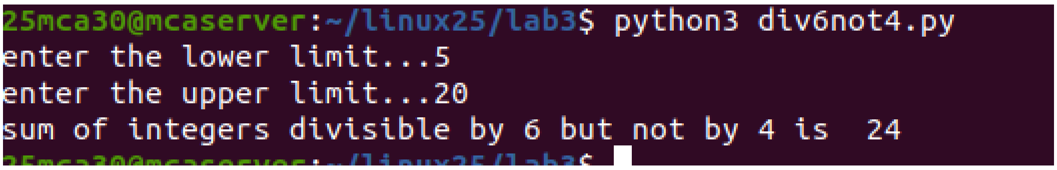
```
print()
```



```
25mca30@mcaserver:~/linux25/lab3$ python3  alter_prime.py
enter a  Number Limit....20
The List of Alternative prime numbers....
2 5 11 17
25mca30@mcaserver:~/linux25/lab3$
```

7. Write a program to compute and display the sum of all integers that are divisible by 6 but not by 4, and that lie below a user-given upper limit.

```
m=int(input("enter        the        lower
limit...")) n=int(input("enter the upper
limit...")) sum=0
for i in range(m,n):
        if   i%6==0   and   i%4!
                =0: sum=sum+i
print("sum of integers divisible by 6 but not by 4 is ",sum)
```

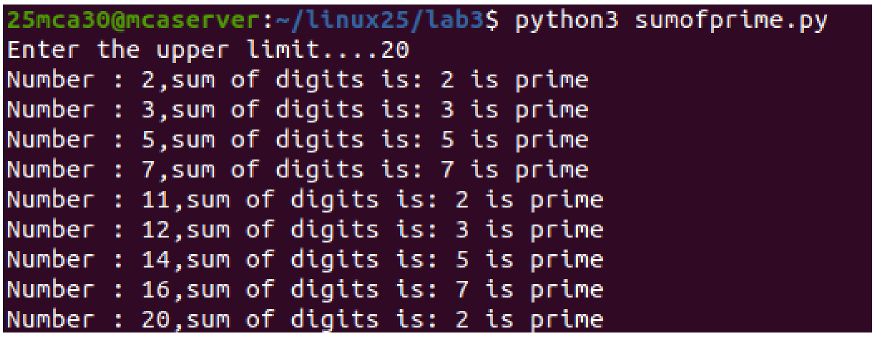


```
25mca30@mcaserver:~/linux25/lab3$ python3 div6not4.py
enter the lower limit...5
enter the upper limit...20
sum of integers divisible by 6 but not by 4 is  24
25mca30@mcaserver:~/linux25/lab3$
```

8. Calculate the sum of the digits of each number within a specified range (from 1 to a user-defined upper limit). Print the sum only if it is prime.

```
def is_prime(num):
        if num<=1:
                return False
        for i in range(2,int(num ** 0.5)+1):
                if num%i==0:
                        return False
        return True
```

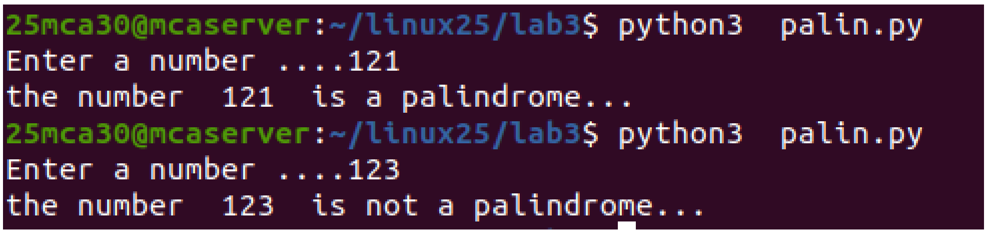
```
n=int(input("Enter the upper limit...."))
for i in range(1,n+1):

    sum_digit=sum(int(j) for j in str(i))
    if is_prime(sum_digit):
        print(f"Number : {i},sum of digits is: {sum_digit} is prime")
```



```
25mca30@mcaserver:~/linux25/lab3$ python3 sumofprime.py
Enter the upper limit....20
Number : 2,sum of digits is: 2 is prime
Number : 3,sum of digits is: 3 is prime
Number : 5,sum of digits is: 5 is prime
Number : 7,sum of digits is: 7 is prime
Number : 11,sum of digits is: 2 is prime
Number : 12,sum of digits is: 3 is prime
Number : 14,sum of digits is: 5 is prime
Number : 16,sum of digits is: 7 is prime
Number : 20,sum of digits is: 2 is prime
```

9. A number is input through the keyboard. Write a program to determine if it’s palindromic.

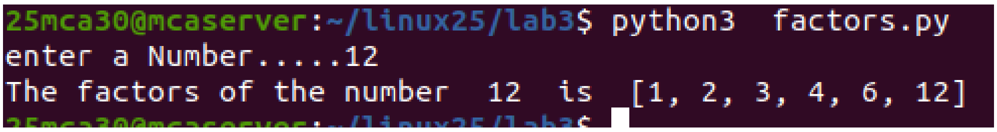
```
n=int(input("Enter a number ...."))

num=n

new_digit=0

while n!=0:

    digit=n%10

    new_digit=new_digit*10+digit

    n=n//10

if num==new_digit:

    print("the number ",num," is a palindrome...")

else:

    print("the number ",num," is not a palindrome...")
```



```
25mca30@mcaserver:~/linux25/lab3$ python3  palin.py
Enter a number ....121
the number  121  is a palindrome...
25mca30@mcaserver:~/linux25/lab3$ python3  palin.py
Enter a number ....123
the number  123  is not a palindrome...
```

10.Write a program to generate all factors of a number. [use while loop]

```
n=int(input("enter a Number…."))
factors=[]
i=1
while i<=n:

    if n%i==0:
        factors.append(i)
    i=i+1

print("The factors of the number ",n," is ",factors)
```

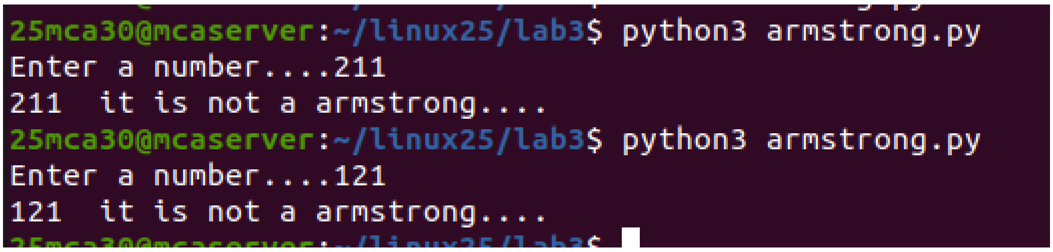


```
25mca30@mcaserver:~/linux25/lab3$ python3  factors.py
enter a Number.....12
The factors of the number  12  is  [1, 2, 3, 4, 6, 12]
25mca30@mcaserver:~/linux25/lab3$
```

11. Write a program to find whether the given number is an Armstrong number or not. [use while loop]

```
num=int(input("Enter a number...."))
number=num
n=len(str(num))
sum=0
while num>0:
    digit=num%10
    sum=sum+digit**n
    num=num//10

if number==sum:
    print(number, " is  a Armstrong number....")
```

```
else:
        print(number," it is not a Armstrong.…")
```



```
25mca30@mcaserver:~/linux25/lab3$ python3 armstrong.py
Enter a number....211
211  it is not a armstrong....
25mca30@mcaserver:~/linux25/lab3$ python3 armstrong.py
Enter a number....121
121  it is not a armstrong....
25mca30@mcaserver:~/linux25/lab3$
```

12. Display the given pyramid with the step number accepted from the user. Eg: N=4

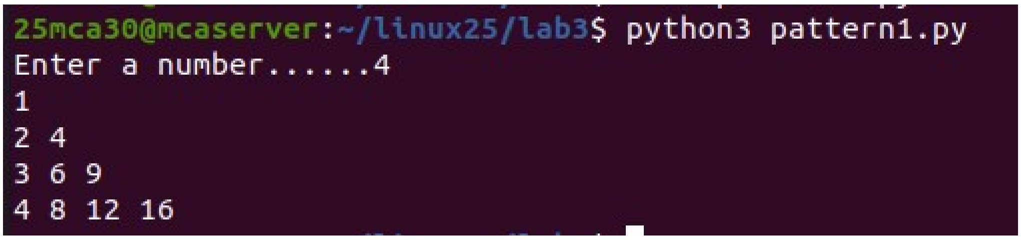
1

2 4

3 6 9

4 8 12 16

```
n=int(input("Enter a number......"))
for i in range(1,n+1):
        for j in range(1,i+1):
                print(j*i,end=" ")
        print()
```

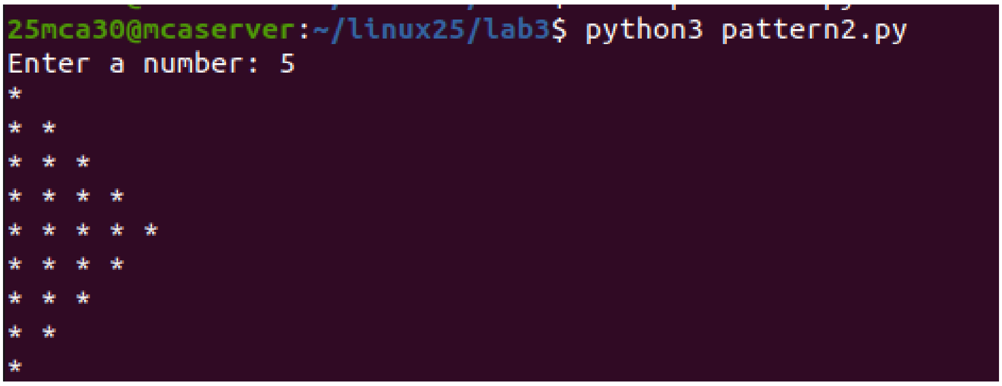


```
25mca30@mcaserver:~/linux25/lab3$ python3 pattern1.py
Enter a number......4
1
2 4
3 6 9
4 8 12 16
```

13. Construct following pattern using nested loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
n = int(input("Enter a number: "))


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


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



```
25mca30@mcaserver:~/linux25/lab3$ python3 pattern2.py
Enter a number: 5
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *
*  *  *
*  *
*
```