Factorial

In [52]:

```
n=int(input("Enter value to find factorial = "))
fact=1
while(n):
    if(n!=0):
        fact*=n
        n-=1
print(fact)
```

```
Enter value to find factorial = 5
120
```

Prime numbers

In [87]:

```
n=1
while(n<20):
    for i in range(1,n):
        if(n%i==0 and i!=1):
            break
        elif(i==n-1):
            print(n)
    n+=1
```

```
2
3
5
7
11
13
17
19
```

Sum of all digits

In [99]:

```
n=(input("Enter the numbers u want to find the sum of = "))
tot=0
for item in n:
    tot+=int(item)
print(tot)
```

```
Enter the numbers u want to find the sum of = 12345
15
```

Fibonacci series

In [108]:

```
n=int(input("Enter range of fibonnaci = "))
i=1
j=2
print(i)
print(j)
count=0
while(count<n):
    third=i+j
    print(third)
    i=j
    j=third
    count+=1
```

```
Enter range of fibonnaci = 10
1
2
3
5
8
13
21
34
55
89
144
233
```

Adding two objects

In [113]:

```
firstname=("Praveen")
secondname=("Kumar")
print("before",id(firstname),id(secondname))
name=firstname+secondname
print(name)
print("after",id(firstname),id(secondname))

"""There is no changes in the id's of the variables this is because
we are making no changes to the variables and the data type of these variables
are tuples which are immutable."""
```

```
before 2778693223792 2778695006832
PraveenKumar
after 2778693223792 2778695006832
```

Out[113]:

```
"As we can see there is no changes in the id's of the variables this is because\nwe are making no changes to the variables
and the data type of these variables \nare tuples"
```

Extending set

In [121]:

```
m={10,20,30,40,50}
n={30,40,50,60,70}
print(m.union(n))
"""From above output it is observable that set data structure does not allow
duplicates which are filtered and only one of each value is appended."""
```

```
{70, 40, 10, 50, 20, 60, 30}
```

Out[121]:

```
'From above output it is observable that set data structure does not allow\nduplicates which are filtered and only one of e
ach value is appended.'
```

Iterations over data structures

In [123]:

```
port1={"FTP":1,"SSH":2,"telenet":3,"http":4}
for item in port1:
    if(port1[item]%2==0):
        print(item)
```

```
SSH
http
```

Iterations over data structures (2)

In [126]:

```
start=1
end=15
dict={}
while(start<end):
    dict[start]=start**2
    start+=1

dict[100]=10000
print(dict)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 100: 10000}
```

Iterations over data structures (3)

In [129]:

```
port2={}
port1={"FTP":21,"SSH":22,"telenet":23,"http":80}
for key in port1:
    port2[port1[key]]=key

print(port1)
print(port2)
```

```
{'FTP': 21, 'SSH': 22, 'telenet': 23, 'http': 80}
{21: 'FTP', 22: 'SSH', 23: 'telenet', 80: 'http'}
```

Iterations over data structures (4)

In [149]:

```
Employees={"Emp1":{"name":"Sara","Dept":"IT","Designation":"Team Lead"},
 "Emp2":{"name":"Anna","Dept":"IT","Designation":"Senior Software Engineer"},
 "Emp3":{"name":"Andy","Dept":"BioTech","Designation":"Senior Software Engineer"},
 "Emp4":{"name":"Andy","Dept":"BioTech","Designation":"Senior Software Engineer"}}
```

In [154]:

```
Master={}
entries=[]
for employee in Employees:
    if(Employees[employee] not in entries):
        Master[employee]=Employees[employee]
        entries.append(Employees[employee])
print(Master)
```

```
{'Emp1': {'name': 'Sara', 'Dept': 'IT', 'Designation': 'Team Lead'}, 'Emp2': {'name': 'Anna', 'Dept': 'IT', 'Designation':
'Senior Software Engineer'}, 'Emp3': {'name': 'Andy', 'Dept': 'BioTech', 'Designation': 'Senior Software Engineer'}}
```