

Even num

In [6]:

```
1 def even(n):
2     if n%2==0:
3         return True
4     else:
5         return False
6 print(even(2))
```

True

In [2]:

```
1 def even(num):
2     if num%2==0:
3         return True
4     return False
5 print(even(2))
```

True

In [3]:

```
1 def even(n):
2     return n%2==0
3 print(even(2))
```

True

In [4]:

```
1 def even(n):
2     return n%2==0
3 for i in range(1,11):
4     if (even(i)):
5         print(i)
```

2
4
6
8
10

In [5]:

```
1 def even(n):
2     return n%2==0
3 c=0
4 while c!=5:
5     if even(i):
6         c+=1
7 print(i)
```

10

Prime num

In [6]:

```
1 def prime(n):
2     count=0
3     for i in range(1,n+1):
4         if n%i==0:
5             count+=1
6     if count==2:
7         return True
8     return False
9 print(prime(3))
```

True

In [7]:

```
1 def prime(num):
2     count=0
3     for i in range(1,num+1):
4         if num%i==0:
5             count+=1
6     return count==2
7 for i in range(40,81):
8     if (prime(i)):
9         print(i,end=' ')
```

41 43 47 53 59 61 67 71 73 79

In [6]:

```
1 def prime(n):
2     if n>1:
3         for i in range(2,n//2+1):
4             if n%i==0:
5                 break
6             else:
7                 return True
8     return False
9 for i in range(1,19):
10     if (prime(i)):
11         print(i,end=' ')
```

2 3 5 7 11 13 17

In [1]:

```
1 def prime(m):
2     if m>1:
3         for i in range(2,m//2+1):
4             if m%i==0:
5                 break
6             else:
7                 return True
8 c=0
9 i=1
10 while c!=6:
11     if (prime(i)):
12         c+=1
13     i+=1
14 i-=1
15 print(i)
```

13

In [17]:

```
1 def prime(N,i):
2     if i>N//2+1:
3         return 0
4     return 1 if N%i==0 else 0+prime(N,i+1)
5 N =9
6 i=2
7 print(prime(N,i))
```

1

Composite num

In [7]:

```
1 # composite num
2 def composite(n):
3     if n>1:
4         for i in range(2,n//2+1):
5             if n%i==0:
6                 return True
7             return False
8 print(composite(4))
```

True

In [11]:

```
1 def composite (n):
2     if n>1:
3         for i in range(2,n//2+1):
4             if n%i==0:
5                 return True
6                 return False
7 for i in range(1,11):
8     if (composite(i)):
9         print(i)
```

4
6
8
10

In [12]:

```
1 def composite(n):
2     if n>1:
3         for i in range(2,n//2+1):
4             if n%i==0:
5                 return True
6                 return False
7 c=0
8 x=1
9 while c!=5:
10     if composite(x):
11         c+=1
12     x+=1
13 print(x-1)
```

12

In []:

1

Niven num

In [14]:

```
1 def niven(num,copy):
2     add=0
3     while num!=0:
4         add+=(num%10)
5         num//=10
6     return copy%add==0
7 n=81
8 if niven(n,n):
9     print('niven num')
10 else:
11     print('not niven')
```

niven num

In [15]:

```
1 def niven(n):
2     add = 0
3     while ( n!=0 ):
4         add += ( n%10 )
5         n //= 10
6     return add
7 X =81
8 if X %(niven(X))==0:
9     print('Niven num')
10 else:
11     print('Not niven num')
```

Niven num

In [16]:

```
1 def niven(n):
2     Add=0
3     Copy = n
4     while n!=0:
5         Add+=(n%10)
6         n //= 10
7     return Copy%Add==0
8 for i in range(1,6):
9     if niven(i):
10        print(i)
```

1
2
3
4
5

In [17]:

```
1 def niven(N):
2     Add = 0
3     Copy = N
4     while N != 0:
5         Add += (N%10)
6         N //= 10
7     return Copy % Add==0
8 c=0
9 i=1
10 while c!=5:
11     if niven(i):
12         c+=1
13     i+=1
14 print(i-1)
```

5

Armstrong num

In [18]:

```
1 def armstrong(n):
2     Add = 0
3     p = len(str(n))
4     while (n!=0):
5         Add += (n%10)**p
6         n //= 10
7     return Add
8 x=153
9 if (x == (armstrong(x))):
10     print('Armstrong num')
11 else:
12     print('Not Armstrong num')
```

Armstrong num

In [19]:

```
1 def Armstrong(Num,p,copy):
2     Add = 0
3     while (Num!=0):
4         Add += (Num%10) ** p
5         Num //= 10
6     return Add == copy
7
8 abc=153
9 if Armstrong(abc,len(str(abc)),abc):
10     print('Armstrong num')
11 else:
12     print('Not Armstrong Num')
```

Armstrong num

In [20]:

```
1 def Armstrong(N):
2     Add = 0
3     P = len(str(N))
4     Copy = N
5     while N != 0:
6         Add += (N%10) ** P
7         N //= 10
8     return Add == Copy
9 for i in range(1,200):
10     if Armstrong(i):
11         print(i)
```

```
1
2
3
4
5
6
7
8
9
153
```

In [21]:

```
1 def Armstrong(Num):
2     Add = 0
3     Copy = Num
4     P = len(str(Num))
5     while (Num!=0):
6         Add += (Num%10) **P
7         Num //= 10
8     return Add == Copy
9 c=0
10 k=1
11 while c != 10:
12     if Armstrong(k):
13         c+=1
14     k+=1
15 print(k-1)
```

153

Disarium Num

In [22]:

```
1 def Disarium(num):
2     Add = 0
3     while num != 0:
4         P = len(str(num))
5         Add += (num%10)** P
6         num //= 10
7     return Add
8 N=135
9 if N == Disarium(N):
10     print('Disarium num')
11 else:
12     print('Not disarium')
```

Disarium num

In [23]:

```
1 def Disarium(Num,P,Copy):
2     Add = 0
3     while Num!=0:
4         Add += (Num%10) ** P
5         Num //= 10
6         P-=1
7     return Add == Copy
8 N=135
9 if (Disarium(N,len(str(N)),N)):
10     print('Disarium Num')
11 else:
12     print('Not Disarium')
```

Disarium Num

In [24]:

```
1 def Disarium(Num):
2     Add = 0
3     Copy = Num
4     while Num != 0:
5         Add += (Num%10) ** (len(str(Num)))
6         Num //= 10
7     return Add == Copy
8 for k in range(1,150):
9     if Disarium(k):
10        print(k)
```

```
1
2
3
4
5
6
7
8
9
89
135
```

In [25]:

```
1 def Disarium(N):
2     Add = 0
3     Copy = N
4     P = len(str(N))
5     while N!=0:
6         Add += (N%10)**P
7         N //=10
8         P-=1
9     return Add == Copy
10 n=1
11 Count=0
12 while Count!=11:
13     if Disarium(n):
14         Count+=1
15     n+=1
16 print(n-1)
```

135

Spy Num

In [26]:

```
1 def Add(Num):
2     Add = 0
3     while Num!=0:
4         Add+=Num%10
5         Num//=10
6     return Add
7 def Pro(Num):
8     Pro=1
9     while Num!=0:
10        Pro*=(Num%10)
11        Num //= 10
12    return Pro
13 N=123
14 if Add(N)== Pro(N):
15     print('Spy num')
16 else:
17     print('Not spy num')
```

Spy num

In [27]:

```
1 def Spy(Num):
2     Add = 0
3     Pro =1
4     while Num!=0:
5         Add+= Num%10
6         Pro*= Num%10
7         Num //= 10
8     return Add == Pro
9 N=123
10 if Spy(N):
11     print('spy num')
12 else:
13     print('Not Spy Num')
```

spy num

In [28]:

```
1 def Spy(N):
2     Add = 0
3     Pro = 1
4     while (N!=0):
5         Add += N%10
6         Pro *= (N%10)
7         N //=10
8     return Add == Pro
9 for m in range(1,50):
10     if Spy(m):
11         print(m)
```

```
1
2
3
4
5
6
7
8
9
22
```


In [29]:

```
1 def Spy(Num):
2     Add = 0
3     Pro = 1
4     while (Num != 0):
5         Rem = (Num%10)
6         Add += Rem
7         Pro *= Rem
8         Num //= 10
9     return Add == Pro
10 Count = 0
11 num = 1
12 while Count != 10:
13     if Spy(num):
14         Count += 1
15     num+=1
16 print(num-1)
```

22

Strong num

In [30]:

```
1 def strongnum(Num,Copy):
2     Add = 0
3     while Num!=0:
4         Fact = 1
5         Rem = (Num %10)
6         for i in range(1,Rem+1):
7             Fact *= i
8         Add += Fact
9         Num //= 10
10    return Add == Copy
11 num=145
12 if strongnum(num,num):
13     print('Strong num')
14 else:
15     print('Not Strong Num')
```

Strong num

In [31]:

```
1 def StrongNum(N):
2     Add = 0
3     Copy = N
4     while (N!=0):
5         Fact = 1
6         Rem = (N%10)
7         for i in range(1,Rem+1):
8             Fact *= i
9         Add += Fact
10        N //= 10
11    return Add == Copy
12 for i in range(1,150):
13     if StrongNum(i):
14         print(i)
```

```
1
2
145
```

In [32]:

```
1 def Strongnum(Num):
2     Add = 0
3     Copy = Num
4     while (Num!=0):
5         Fact = 1
6         Rem = (Num%10)
7         for i in range(1,Rem+1):
8             Fact *= i
9         Num //= 10
10        Add += Fact
11    if Add == Copy:
12        return True
13    else:
14        return False
15 C = 0
16 N = 1
17 while (C!=3):
18     if Strongnum(N):
19         C += 1
20     N += 1
21 print(N-1)
```

145

EMIRP Num

In [33]:

```
1 def Polindrome(Num):
2     Rev = 0
3     Copy = Num
4     while (Num!=0):
5         Rev = Rev*10 + (Num%10)
6         Num //= 10
7     return Rev
8 def Prime(Num):
9     if Num>1:
10         for i in range(2,Num//2+1):
11             if Num % i==0:
12                 break
13         else:
14             return True
15 N=13
16 Rev = Polindrome(N)
17 if N != Rev and Prime(N) and Prime(Rev):
18     print('EMIRP Num')
19 else:
20     print('Not EMIRP Num')
```

EMIRP Num

In [34]:

```
1 def EMIRP(Num):
2     Rev = 0
3     copy = Num
4     while Num!=0:
5         Rev = Rev*10 + (Num%10)
6         Num //= 10
7     if Rev != copy and copy>1:
8         for i in range(2,copy//2+1):
9             if copy%i==0:
10                break
11        else:
12            if Rev>1:
13                for k in range(2,Rev//2+1):
14                    if Rev%k==0:
15                        break
16                return True
17            return False
18    return False
19 N=13
20 if EMIRP(N):
21     print('EMIRP Num')
22 else:
23     print('Not EMIRP Num')
```

EMIRP Num

In [39]:

```
1 def emirp (Num):
2     Rev = 0
3     Copy = Num
4     while Num !=0:
5         Rev = Rev*10+(Num%10)
6         Num //= 10
7     if Rev != Copy and Copy>1:
8         for i in range(2,Copy//2+1):
9             if Copy%i==0:
10                break
11        else:
12            if Rev>1:
13                for k in range(2,Rev//2+1):
14                    if Rev%k==0:
15                        return False
16                else:
17                    return True
18            return False
19    return False
20 for i in range(1,50):
21     if emirp(i):
22         print(i)
```

13

17

31

37

In [47]:

```
1 def Emirp(num) :
2     Rev = 0
3     while num !=0:
4         Rev = Rev*10+(num%10)
5         num //= 10
6     return Rev
7 def prime(num):
8     if num>1:
9         for i in range(2,num//2+1):
10             if num%i==0:
11                 return False
12             else:
13                 return True
14     return False
15 Count = 0
16 I = 1
17 while Count!=4:
18     if I != Emirp(I)and prime(I) and prime(Emirp(I)):
19         Count+=1
20     I += 1
21 print(I-1)
```

37

Happy Num

In [51]:

```
1 def Sumsq(Num):
2     Sq = 0
3     while (Num!=0):
4         Sq+= (Num%10)**2
5         Num //= 10
6     return Sq
7 def Happynum(Num):
8     while Num>9:
9         Num = Sumsq(Num)
10    return True
11 I=49
12 if Happynum(I):
13     print ('Happy Num')
14 else:
15     print('Not Happy')
```

Happy Num

In [55]:

```
1 def Happynum(Num):
2     while Num>9:
3         Sq = 0
4         while (Num!=0):
5             Sq += (Num%10)**2
6             Num //= 10
7         Num=Sq
8     return Num == 1
9 N = 48
10 if Happynum(N):
11     print('Happy number')
12 else:
13     print('Not happy num')
```

Not happy num

In [53]:

```
1 def Happynum(Num):
2     while Num>9:
3         Sq = 0
4         while (Num!=0):
5             Sq += (Num%10)**2
6             Num //= 10
7         Num=Sq
8     return Num == 1
9 for i in range(1,50):
10     if Happynum(i):
11         print(i)
```

```
1
10
13
19
23
28
31
32
44
49
```

In [3]:

```
1 def SqNum(Num):
2     Sq = 0
3     while Num != 0:
4         Sq += (Num%10)**2
5         Num //= 10
6     return Sq
7 def HappyNum(Num):
8     while Num>9:
9         Num=SqNum(Num)
10    return Num==1
11 n=1
12 count=0
13 while count!=5:
14     if HappyNum(n):
15         count+=1
16     n+=1
17 print(n-1)
```

23

Sunny Number

In [3]:

```
1 def SunnyNum(Num):
2     N = 1
3     while N<Num:
4         if N*N==Num+1:
5             return True
6         N+= 1
7     return False
8 N=8
9 if SunnyNum(N):
10     print('Sunny Num')
11 else:
12     print('Not Sunny')
```

Sunny Num

In [7]:

```
1 def Sunny(Num):
2     N = 1
3     while (N<Num):
4         if N**2 == Num+1:
5             return 'Sunny Num'
6         N += 1
7     return 'Not sunny'
8 Sunny(8)
```

Out[7]:

'Sunny Num'

In [8]:

```
1 def Sunny(Num):
2     N = 1
3     while N<Num:
4         if N**2 == Num+1:
5             return True
6         N += 1
7     return False
8 for i in range(1,50):
9     if Sunny(i):
10        print(i)
```

3
8
15
24
35
48

In [9]:

```
1 def Sunny(Num):
2     I = 1
3     while I<Num:
4         if I**2 == Num+1:
5             return True
6         I += 1
7     return False
8 Count = 0
9 N = 1
10 while Count != 5:
11     if Sunny(N):
12         Count += 1
13     N += 1
14 print(N-1)
```

35

Automorphic Num

In [12]:

```
1 def Auto(Num):
2     Sq = Num**2
3     while Num!=0:
4         if (Num%10) != (Sq%10):
5             return False
6         Num //= 10
7         Sq //= 10
8     return True
9 N = 25
10 if Auto(N):
11     print ('Automorphic Num')
12 else:
13     print('Not Automorphic Num')
```

Automorphic Num

In [13]:

```
1 def Auto(N):
2     Sq = N**2
3     while (N!=0):
4         if ((N%10) != (Sq%10)):
5             return False
6         N //= 10
7         Sq //= 10
8     return True
9 for i in range(1,50):
10     if Auto(i):
11         print(i)
```

1
5
6
25

In [14]:

```
1 def Auto(N):
2     Sq = N**2
3     while (N!=0):
4         if ((N%10) != (Sq%10)):
5             return False
6         N //= 10
7         Sq //= 10
8     return True
9 Count = 0
10 N = 1
11 while Count != 4:
12     if Auto(N):
13         Count += 1
14     N += 1
15 print(N-1)
```

25

Trimorphic Number

In [15]:

```
1 def Trimorphic(Num):
2     Cube = Num**3
3     while Num!=0:
4         if (Num%10)!= (Cube%10):
5             return False
6         Num //= 10
7         Cube //= 10
8     return True
9 N = 25
10 if Trimorphic(N):
11     print('Trimorphic Num')
12 else:
13     print('Not Trimorphic')
```

Trimorphic Num

In [17]:

```
1 def Trimorphic(Num):
2     Cube = Num**3
3     while Num!=0:
4         if (Num%10)!= (Cube%10):
5             return False
6             Num //= 10
7             Cube //= 10
8     return True
9 for i in range(1,100):
10     if Trimorphic(i):
11         print(i)
```

```
1
4
5
6
9
24
25
49
51
75
76
99
```


In [18]:

```
1 def Trimorphic(Num):
2     Cube = Num**3
3     while Num!=0:
4         if (Num%10)!= (Cube%10):
5             return False
6         Num //= 10
7         Cube //= 10
8     return True
9 C = 0
10 N = 1
11 while C!= 10:
12     if Trimorphic(N):
13         C += 1
14     N += 1
15 print(N-1)
```

75

Perfect Number

In [6]:

```
1 def Perfect(Num):
2     M = 0
3     for i in range(1,Num//2+1):
4         if Num%i == 0:
5             M+=i
6     if Num == M:
7         return True
8     else:
9         return False
10 N = 6
11 if Perfect(N):
12     print('Perfect Num')
13 else:
14     print('Not perfect Num')
```

Not perfect Num

In [3]:

```
1 def Perfect(Num):
2     M = 0
3     for i in range(1,Num//2+1):
4         if Num % i == 0:
5             M += i
6     return Num == M
7 for i in range(1,1000):
8     if Perfect(i):
9         print(i)
```

6
28
496
8128

In [28]:

```
1 def Perfect(Num):
2     M = 0
3     for i in range(1,Num//2+1):
4         if Num % i == 0:
5             M +=i
6     return Num == M
7 C = 0
8 N = 1
9 while C!=5:
10     if Perfect(N):
11         C += 1
12     N += 1
13 print(N-1)
```

14

Neon Number

In [29]:

```
1 def Neon(Num):
2     Sq = Num*Num
3     Add = 0
4     while Sq != 0:
5         Add += (Sq%10)
6         Sq //= 10
7     return Add == Num
8 N = 9
9 if Neon(N):
10     print('Neon num')
11 else:
12     print('Not Neon')
```

Neon num

In [30]:

```
1 def Neon(Num):
2     Sq = Num*Num
3     Add = 0
4     while Sq != 0:
5         Add += (Sq%10)
6         Sq //= 10
7     return Add == Num
8 for i in range(1,50):
9     if Neon(i):
10        print(i)
11
```

1
9

In [34]:

```
1 def Neon(Num):
2     Sq = Num*Num
3     Add = 0
4     while Sq != 0:
5         Add += (Sq%10)
6         Sq //= 10
7     return Add == Num
8 C = 0
9 N = 1
10 while (C!= 2):
11     if Neon(N):
12         C+= 1
13         N+= 1
14 print(N-1)
```

9

In [21]:

```

1  def Star(N):
2      St = 1
3      Sp = N-1
4      for i in range(N):
5          for k in range(Sp):
6              print(' ',end=' ')
7          for s in range(St):
8              print('*',end=' ')
9          print()
10         if i< N//2:
11             St += 2
12             Sp -= 1
13         else:
14             St -= 2
15             Sp += 1
16 N = int(input())
17 Star(N)

```

9

```

          *
        * * *
      * * * * *
    * * * * * * *
  * * * * * * * *
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *

```

In [41]:

```

1  def Num(N):
2      St = 1
3      Sp = N-1
4      if N%2==0:
5          N+=1
6      for A in range(N):
7          Num=St//2+1
8          for B in range(Sp):
9              print(' ',end=' ')
10         for C in range(St):
11             print(Num,end=' ')
12             if C<St//2:
13                 Num += 1
14             else:
15                 Num -= 1
16         print()
17         if A< N//2:
18             St +=2
19             Sp -=1
20         else:
21             St -=2
22             Sp +=1
23 N = int(input())
24 Num(N)

```

7

```

      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
  3 4 5 4 3
    2 3 2
      1

```

Pronic Number

In [30]:

```
1 def Pronic(Num):
2     i = 1
3     while i<Num:
4         if i*(i+1)==Num:
5             return True
6         i+=1
7     return False
8 N = 6
9 if Pronic(N):
10     print('Pronic num')
11 else:
12     print('Not pronic')
```

Pronic num

In [31]:

```
1 def Pronic(Num):
2     i = 1
3     while i<Num:
4         if i*(i+1)==Num:
5             return True
6         i+=1
7     return False
8 for i in range(1,50):
9     if Pronic(i):
10         print(i)
```

```
2
6
12
20
30
42
```

In [32]:

```
1 def Pronic(Num):
2     i = 1
3     while i<Num:
4         if i*(i+1)==Num:
5             return True
6         i+=1
7     return False
8 C = 0
9 N = 1
10 while C != 5:
11     if Pronic(N):
12         C +=1
13     N += 1
14 print(N-1)
```

30

Dec to Binary

In [34]:

```
1 def Binary(Num):
2     Add = 0
3     X = 1
4     while Num != 0:
5         Add +=(Num%2)*X
6         Num //= 2
7         X *= 10
8     return Add
9 N = 12
10 print(Binary(N))
```

1100

Binary to Dec

In [35]:

```
1 def Decimal(Num):
2     Add = 0
3     X = 1
4     while Num!=0:
5         Add +=(Num%10)*X
6         Num //= 10
7         X *= 2
8     return Add
9 N = 1100
10 print(Decimal(N))
```

12

Polindrome

In [37]:

```
1 def Rev(Num):
2     Add = 0
3     while Num!=0:
4         Add=Add*10+(Num%10)
5         Num //= 10
6     return Add
7 N = 121
8 print(Rev(N)==N)
```

True

Eval Number

In [38]:

```
1 def Eval(Num):
2     C = 0
3     while Num != 0:
4         if Num%2 ==1:
5             C +=1
6             Num //=2
7     return C%2==0
8 N = 12
9 if Eval(N):
10     print('Eval Num')
11 else:
12     print('Odulous Num')
```

Eval Num

In [4]:

```
1 def pattern(n,sp,st):
2     d=65
3     for a in range(n):
4         for b in range(sp):
5             print(' ',end=' ')
6         for c in range(st):
7             print(chr(d),end=' ')
8             d=d+1
9         print()
10        if a<n//2:
11            st+=2
12            sp-=1
13        else:
14            st-=2
15            sp+=1
16    pattern(7,7//2,1)
```

```
      A
     B C D
    E F G H I
   J K L M N O P
  Q R S T U
 V W X
 Y
```

In [20]:

```

1  def pattern(N,Sp,St):
2      if N%2==0:
3          N-=1
4      A=65
5      for a in range(N):
6          for b in range(Sp):
7              print(' ',end=' ')
8          for c in range(St):
9              print(chr(A),end=' ')
10         A+=1
11         print()
12         if a<N//2:
13             St-=2
14             Sp+=1
15
16         else:
17             St+=2
18             Sp-=1
19     N=9
20     pattern(N,0,N)

```

```

A B C D E F G H I
  J K L M N O P
    Q R S T U
      V W X
        Y
          Z [ \
            ] ^ _ ` a
              b c d e f g h
                i j k l m n o p q

```

In []:

1

In []:

1