① Armstrong number ⟶

```
a = int (input ('Enter the number to be checked : '))
temp = a
Sum = 0
Count = 0
while (temp > 0) :
        temp // = 10
        count + = 1
temp = a
while (a > 0) :
        digit = a % 10
        sum + = digit ** count
        a // = 10
if (sum == temp) :
        print ('The numbes is an Armstrong numbes')
else :
        print ('The numbes is not an Armstrong numbes')
```

(153 = $1^3 + 5^3 + 3^3$)

No. of digits की power की sum

② Armstrong number in range ⟶

```
lower = int (input ('Enter the lower limit : '))
upper = ———————————— upper ————
for num in range (lower, upper + 1 ) :
        Sum = 0
        temp = num
        count = 0
        while (temp > 0) :
                temp // = 10
                count + = 1
        temp = num
        while (num > 0) :
                digit = num % 10
                sum + = digit ** count
                num // = 10
        if (sum == temp) :
        print (f' {temp} is an
        armstrong number')
```

Scanned by TapScanner

③ Balanced unbalanced list ⟶

```
c=0
l=input("enter the element of list separated by spaces =")
l = l.split(" ")
sl = 0
for i in range (len(l)):
        sl = sl + int (l[i])
f=0
for i in range (len(l)-1):
        f = f + int(l[i])
        sl = sl - int(l[i])
        if (f==sl):
                c=c+1
                break
    if (c==0):
            print ("not balanced list")
    else:
            print ("balanced list")
```

④ Clockwise inverted pyramid

```
rows = int (input('Enter the number of rows: '))
for i in range (0,rows):
        for j in range (0, i+1):
            print ("*", end = ' ')
        print ("\r")

for i in range (rows, 0, -1):
        for j in range (0, i-1):
            print ("*", end = ' ')
        print ('\r')
```

Scanned by TapScanner

⑤ Program to count the frequency of words in a giuen String. ⟶

```
a = input ("Enter the String : ")
frequency_ counter = { }
for i in a.split ( ) :
    keys = frequency — counter. keys ()
    if i in keys:
        frequency — counter [i] += 1
    else :
        frequency_ counter [i] = 1
print (frequency counter)
```

⑥ counting the frequency of character in a given String ⟶

```
a = input ("Enter the String : ")
b = input ("Enter the characters to be counted : ")
print (a. count (b))
```

counting the frequency of all characters in a String

```
a = input ("Enter the String :")
frequency counter = [ ]
for i in a:
    keys = frequency — counter. keys ()
if i in keys :
    frequency — counter [i] + = 1
else :
    frequency — counter [i] = 1
print (frequency — counter)
```

⑦ creating String ⟶

```
a = " This is a String"
print (a)
print (type (a))
```

Scanned by TapScanner

⑧ downward full pyramid

```
rows = int (input ())
col = 2* rows - 2
for i in range (rows, -1, -1):
        for j in range (col, 0, -1):
            print (end = "  ")
        col = 1
        for j in range (0, i+1):
            print ("*", end = "  ")
        print ("  ")
```

⑨ downward half pyramid

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

⑩ To find factorial ⟶

```
a = int (input ('Enter the no. whose factorial to be found : '))
b = a
fact = 1
while (a>0):
    fact = fact * a
    a- = 1
print (f' The factorial of {b} is : {fact}')
```

⑪ Fibonacci series ⟶

```
n = int (input ('Enter the no. of terms to be displayed'))
first = 0
second = 1
```

Scanned by TapScanner

```
if n==1:
    print (f' The fibonacci series for {n} term is : {first}')
elif n==2:
    print (f' The fibonacci series for {n} term is : {first, second}')
elif n==0:
    print (f' The fibonacci series has no terms ')
else:
    print (f' The fibonacci series is for {n} term is : \n {first}, {
    second}, 'end = ' ')
while (n>2):
        third = first + second
        print (f' {third}, 'end= '')
        first = second
        second = third
        n-=1
```

(12) first last swap

```
a= input ("Enter the String : ")
print (f' The String before swapping is : {a})
 a= a[-1] + a [1:-1] + a [0]
printf' The String after swapping is : {a}')
```

(13) Full triangle

```
rows = int (input ("Enter the no. of rows :"))
col= 2* rows - 2
for i in range (0, rows , 1):
        for j in range (0, col, 1):
            print (end = "  ")
    col- = 1
    for j in range (0, i+1):
            print ("*", end = "  ")
    print(" ")
```

Scanned by TapScanner

(14) HCF, LCM ⟶

```
a = int (input ("Enter the first number : "))
b = int (input ("Enter the second number : "))
if a>b :
        small =b
else :
        small =a
for i in range (small, 0, -1):
        if a%.i ==0  and b%.i ==0 :
                hcf = i
                break
lcm = (a* b)/hcf
print (f' The hcf of {a} and {b} is hcf In The LCM of {a}
        and {b} is{int (lcm)}')
```

(15) Join

```
tu = ('farman', 'gla', 'python')
x = '@'.join (tu)
print (x)
```

output

farman @ gla @ python

(16) Length of a String

```
a = input ('Enter a string :')
print (f' The length of the string is : {len (a)}')
count =0
for i in a :
    count = count +1
print(f' The length of the string is {count}')
```

(7) To print minor Δ right angle

```
row = int (input ( " Enter the no. of rows" ))
k = (2* row) -2
for i in range (0, row):
        for j in range (0,k):
```

Scanned by TapScanner

```
        print (end = ' ')
    k = k-2
    for j in range (0, i+1):
        print ("*", end = ' ')
    print ('\r')
```

(18) # program for removing character from a given string ⟶

```
a = input ('Enter the String : ')
b = input ('Enter the characters to be removed : ')
a = a.replace (b, '', 1)
print()
```

(19) program for removing character of given index ⟶

```
a = input ('Enter the String : ')
b = int (input ('Enter the index of the character to be removed : '))
a = a [:b] + a[b+1 :]
print (a)
```

(20) Removing duplicate in list ⟶

```
original = eval (input ('Enter the element : '))
new = [ ]
for i in original :
        if i not in new :
                new.append (i)
print (new)
```

(21) right angle triangle

```
row = int (input (' Enter the number of rows : '))
for i in range (0, row):
    for j in range (0, i):
        print ("*", end = ' ')
    print ('\n')
```

Scanned by TapScanner

(22) Strong number (sum of factorial of digit of a number) ⟶

```
n = int (input())
m = n
Sum = 0
while (m > 0):
    rem = m % 10
    fact = 1
    while (rem > 0):
    fact   fact * = rem
           rem - = 1
    Sum = sum + fact
    m // = 10
if sum = = n:
    print (f' {n} is a strong number')
else:
    print (f' {n} is not a strong number')
```

(23) Sum of natural numbers ⟶

```
n = int (input ('Enter the upper limit of Sum : '))
Sum = 0
for i in range (1, n+1):
    sum + = i
print (f' The Sum of all natural number upto {n} is {sum}')
```

(24) Swapping character of a given String Character

```
a = input ('Enter the string : ')
b = input ('Enter the characters to be replaced : ')
c = input ('Enter the characters to be replaced with : ')
a = a.replace (b, c)
print (a)
```

Scanned by TapScanner

(25) Swapping two numbers using a temporary variable

```
a = int (input ('Enter the first number : '))
b = int (input ('Enter the second number : '))
temp = a
  a = b
  b = temp
print ('The first number is now : ', a)
print ('The second number is now : ', b)
```

(26) Swapping two numbers without using a temporary variable

```
a = int (input ('Enter the first number : '))
b = int (input ('Enter the second number : '))
a = a+b
b = a-b
a = a-b
print ('The first number is now : ', a)
print ('The second number is now : ', b)
```

(27) Swapping two numbers using Bitwise operator

```
a = int (input ('Enter the first number : '))
b = int (input ('Enter the second number : '))
a = a^b
b = a^b
a = a^b
print ('The first number is now : ', a)
print ('The second number is now : ', b)
```

Scanned by TapScanner

(28) Palindrome number ——→

```
s = input ("Enter the string :")
a = s[::-1]
if (s==a):
    print ("The given string is palindrome")
else :
    print (" The given string is not a palindrome")
```

```
n = int (input ("Enter number :")) ——→
temp = n
rev = 0
while (n>0)
    dig = n % 10
    rev = rev * 10 + dig
    n = n // 10
if (temp = rev):
    print ("The number is a palindrome")
else :
    print ("not palindrome")
```

(29) Prime number ——→

```
num = int (input ("Enter the number"))
flag = false
if num > 1:
    for i in range (2, num):
        if (num % i) == 0:
            flag = true
            break
if flag :
    print ("not a prime number")
else :
    print (" prime number")
```

Scanned by TapScanner

(30) Average of numbers: ⟶

```
n= int (input ("Enter number"))
sum=0
for num in range (1, n+1, 1):
    sum= sum + num
print ("sum of first", n, "numbers is :", sum)
average =sum/n
print ("average of numbers = ", average)
```

(31) String formatting ⟶

a) formatting with % operator
b) formatting with format () String method
c) Formatting with String literals
d) formatting with String template class

(32) Anagram ⟶

```
a = input ()
b = input ()
if sorted (a) == sorted (b):
    print ("anagram")
else:
    print ("not anagram")
```

Scanned by TapScanner