

1) WAP to divide two numbers.

~~$x = \text{int}(\text{input}("enter"))$~~

$n_1 = 5$

$n_2 = 2$

$\text{print}(n_1 // n_2)$

2) WAP to divide two numbers without using division operator.

~~$\text{num} = 15$~~

~~$\text{num1} = 15$~~

~~$\text{div} = 5$~~

~~$\text{div} = 4$~~

~~Count = 0    res = 0~~

~~while    count < div~~

~~Count = 0~~

~~$\text{res} = \text{num} - \text{div}$~~

~~while    num1 >= div :~~

~~Count + = 1~~

~~$\text{num1} = \text{num1} - \text{div}$~~

~~$\text{print}(\text{count})$~~

~~Count + = 1~~

~~$\text{print}(\text{count})$~~

3) WAP to multiply 2 numbers without using multiplication operator.

Tracing

~~$\text{num} = 15$~~

~~$\text{num} = 15$~~

~~$\text{mul} = 3$~~

~~$\text{mul} = 3$~~

~~$\text{out} = 0$~~

~~for i in range(mul):~~

~~(0, 1, 2)~~

~~$\text{out} + = \text{num}$~~

~~$i = 0$~~

~~$\text{print}(\text{out})$~~

~~$\text{out} = 0 + 15 = 15$~~

~~$i = 1$~~

~~$\text{out} = 15 + 15 = 30$~~

~~$i = 2$~~

~~$\text{out} = 30 + 15 = 45$~~

O/p :- 45 //

4) The parameter weekday is true if it is a weekday and the parameter vacation is true if it is a vacation. If we are on a vacation we sleep in if it is not a weekday or we are on a vacation display true when we sleep in else display false.

⇒

weekday = true → weekday  
= failure → not a weekday

vacation = true → on a vacation  
= False → not on vacation

weekday == False or vacation == True → Sleeping program.

if weekday = ~~False~~ = False or vacation = True  
print ("Sleeping")  
else:  
print ("not sleeping")

5) We have two monkeys A and B and the parameters ASmile and BSmile indicate if each is smiling. We are in trouble if they both are smiling or if neither of one is smiling display true if we are in trouble else false.

⇒

ASmile = True  
BSmile = True  
if ( ASmile == True and BSmile == True ) or  
( ASmile == False and BSmile == False ):  
print ('True')  
else:  
print ('False')

a weekday

it is a

cation

y false.

Sleeping

True

parameters.

elling.

ng as

f we are

or

:

6)

Given two integer values a and b display this sum unless the two values are same then display double their sum.

⇒

$a = \text{int}(\text{input}('enter a'))$

$b = \text{int}(\text{input}('enter b'))$

if  $a == b$

print( $2 * (a + b)$ )

else:

print( $a + b$ )

7)

Given an integer n display absolute difference between n and 21 except display double the absolute difference if n is over 21.

⇒

$n = 7$

If  $n < 21$

print( $21 - n$ )

else:

print( $2 * (n - 21)$ )

8)

You are driving too fast and a police officer stops you. write a code to determine the following result encoded as int value.

0 = no ticket

1 = Small ticket

2 = big ticket

If the Speed is 60 or less issue(display) 0

If the Speed is between 61 and 80 inclusive display(1) and if the Speed is 81 or more display 2.

Unless it is your birthday on that day your Speed can be 5 higher in all the cases.

⇒

Speed = 60

0 = no ticket

1 = Small - ticket

```
2 = big-ticket  
birthday = True  
if birthday == False  
if speed <= 60:  
    print(0)  
elif 61 < speed <= 80:  
    print(1)  
else:  
    print(2)  
else:  
    if speed <= 65:  
        print(0)  
    elif 66 < speed <= 85:  
        print(1)  
    else:  
        print(2)
```

(Ans)

Speed = 63  
Birthday = False  
temp = 0

if birthday == True:  
 temp = 5

if speed <= (60 + temp):  
 print('no ticket')

elif speed >= (61 + temp) and speed <= (80 + temp):  
 print("1 small ticket")

else:  
 print("2 big ticket come back again")

9) We have business standing in a line, numbered 1, 2, 3, 4, ... the odd business (1, 3, 5, ...) have the normal two cars the even business (0, 2, 4, ...) have 3 cars display the total number of cars of all the n business standing in the line.



$$\text{Sum} = 0$$

$$n = 10$$

for i in range (1, n+1):

if i%2 == 0:

$$\text{Sum} + = 3$$

else:

$$\text{Sum} + = 2$$

print ('total number of car of all the n business', Sum)

10)

We have triangles made of blocks the top most row has 1 block, the next down has two blocks, the next row 3 blocks and so on determine the total number of blocks in a triangle with n rows.



$$n = 10$$

$$\text{Sum} = 0$$

for i in range (1, n+1):

$$\text{Sum} + = i$$

print (Sum)

y):

25/11/20

1) for i in range(3):  
    for j in range(1, 5, 2):  
        print("hi")

O/P :-

hi  
hi  
hi  
hi  
hi

2) for i in range(5, 0, -3):  
    for j in range(1, 5, 2):  
        print("hi")

O/P :-

hi  
hi  
hi  
hi

3) for rows in range(4):

    for col in range(4):  
        print("\*", end=" ")  
    print()

Tracing :- row = (0, 1, 2, 3)

row = 0

col = (0, 1, 2, 3)

col = 0  $\Rightarrow$  print("\*");

col = 1  $\Rightarrow$  print("\*");

col = 2  $\Rightarrow$  print("\*");

col = 3  $\Rightarrow$  print("\*");

inner loop completed

print()

row = 1

"

$j \neq 2$

"

$i = 3$

"

O/P :-

*	*	*	*
00	01	02	03
*	*	*	*
10	11	12	13
*	*	*	*
20	21	22	23
*	*	*	*
30	31	32	33

4) Write a program to print a below pattern

→ O/P:- \* 00 01 02 03  
                  \* 10 11 12 13  
                  \* 20 21 22 23  
                  \* 30 31 32 33

→ for row in range(4):

    for col in range(4):

        if row == 0 or row == 3 or col == 0 or col == 3:  
             print("\*", end="")

        else:

            print(" ", end="")

    print()

row = (0, 1, 2, 3)

Tracing: row = 0

col = (0, 1, 2, 3)

col = 0

(0r)

n = 15

for row in range(n):

    for col in range(n):

        if row == 0 or row == n-1 or col == 0 or

        col == n-1:

            print("\*", end="")

        else:

            print(" ", end="")

    print()

5) O/P :- \* \* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

$\Rightarrow n=5$

for rows in range(n) :

for col in range(n) :

if rows == 0 or rows == n-1 or col == 0 or  
 col == n-1 or rows == col or  
~~rows < col~~ rows + col == n-1 : ↴

print ("\*", end = " ")

else :

print (" ", end = " ")

print()

6)  $n=7$

for rows in range(n) :

for col in range(n) :

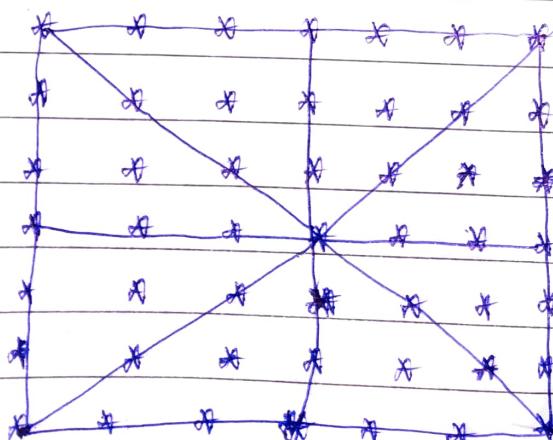
if rows == 0 or rows == n-1 or rows == n//2  
 or col == n//2 or col == n-1 or rows == col  
 or rows + col == n-1 or col == 0 :  
 print ("\*", end = " ")

else :

print (" ", end = " ")

print()

O/P :-



7) O/P :- 1 2 3 4  
 5 6 7 8  
 9 10 11 12  
 13 14 15 16

→ n = 4

Count = 1

for row in range(n):

for col in range(n):

print(count, end="")

Count += 1

print()

8) O/P :- A B C D E  
 F G H  
 I J K L  
 M N O P

→ n = 4

Count = 65

for row in range(n):

for col in range(n):

print(chr(count), end="")

Count += 1

print()

9) O/P :- 1 \* 2 \* 3 \* 4 \* 5  
 11 \* 12 \* 13 \* 14 \* 15  
 21 \* 22 \* 23 \* 24 \* 25  
 16 \* 17 \* 18 \* 19 \* 20  
 6 \* 7 \* 8 \* 9 \* 10

→ n = 5

Count = 1

for row in range(n):

for col in range(n) :

    print(count, end = " ")

    if col != n - 1 :

        print("#", end = " ")

    count += 1

    print()

    if rows < n // 2 :

        count += n

    else:

        count -= (rows \* n)

logic

1 \* 2 \* 3 \* 4 \* 5 → row = 0 → +5

11 \* 12 \* 13 \* 14 \* 15 → row = 1 → +5

21 \* 22 \* 23 \* 24 \* 25 → row = 2 → rows \* 5  
→ -10

16 \* 17 \* 18 \* 19 \* 20 → row = 3 → rows \* 5  
→ -15

6 \* 7 \* 8 \* 9 \* 10

10) O/p :- Aa Bb Cc

    Dd Ee Ff

    Gg Hh Ii

⇒ n = 3

Count = 65

Count2 = 97

for rows in range(n) :

    for col in range(n) :

        print(chr(count), end = " ")

        print(chr(count), end = " ")

    Count += 1

    Count2 += 1

    print()

By using Single for loop

ii) O/p :- \* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

$$\Rightarrow n = 4$$

for i in range(n):  
print("\*" \* n)

12) O/p :- \*

\* \*

\* \* \*

\* \* \* \*

$$\Rightarrow n = 4$$

for row in range(n):

for col in range(n):

if row == col or row > col:  
print("\*", end="")

print()

else:

print(" ", end="")

print()

(or)

$$n = 4$$

for row in range(n):

for col in range(n):

(row+1)

if row >= col:

print("\*", end="")

X

print()

<u>Tracing</u> :-	now = (0, 1, 2, 3)	
	now = 0	now = 2
	col = (0, 1, 2, 3)	2 >= 0 T
	col = 0	2 >= 1 F
	if 0 >= 0 true	2 >= 2 T
	print *	2 >= 3 F
	if 0 >= 1 False	now = 3
	0 >= 2 F	3 >= 0 T
	0 >= 3 F	3 >= 1 T
	now = 1	3 >= 2 T
	1 >= 0 T	3 >= 3 T
	1 >= 1 F	
	1 >= 2 F	
	1 >= 3 F	

13) \* \* \* \*  
       \* \* \*  
       \* \*  
       \*

⇒

$$n = 4$$

for now in range(n):

    for col in range(n):

        if now + col <= n - 1

            print(" \* ", end = " ")

    print()

(0n)

$$n = 4$$

for now in range(n):

    for col in range(n - now):

        print(" \* ", end = " ")

    print()

14) O/P :- \* \* \* \*  
\* \* \*  
\* \*  
\*

n = 4

for row in range(n):

for col in range(n):

if row == col or row < 0 == col :

print ("\*", end = " ")

else:

print (" ", end = " ")

print()

15) O/P :- \*  
\* \*  
\* \* \*  
\* \* \* \*

→ n = 4

for row in range(n):

for col in range(n):

if row + col >= n - 1

print ("\*", end = " ")

print()

else:

print (" ", end = " ")

print()

16) O/P :-

2	3		
4	5	6	
7	8	9	10

→ n = 4

Count = 1

for rows in range(n):

    for col in range(n):

        if rows >= col:

            print ( count, end = " " )

            Count += 1

    print ( )

17) O/P :-

1	2	3	4
5	6	7	

8	9
---	---

10
----

→ n = 4

Count = 1

for rows in range(n):

    for col in range(n):

        if rows <= col:

            print ( count, end = " " )

            Count += 1

    else:

        print ( " " , end = " " )

    print ( )

18) O/p :- 1 2 3 4

5 6 7

8 9

10

 $\Rightarrow n = 4$ 

Count = 1

for row in range(n):

for col in range(n):

if rows + col &lt;= n - 1:

print('count', end = " ")

Count += 1

else:

print(' ', end = " ")

print()

19) O/p :-       1

2 3

4 5 6

7 8 9 10

 $\Rightarrow n = 4$ 

Count = 1

for row in range(n):

for col in range(n):

if rows + col &gt;= n - 1:

print('count', end = " ")

Count += 1

else:

print(' ', end = " ")

print()

(20) Output : 1

	3	2		
	6	5	4	
	10	9	2	7



Count = 1

temp = 5

for i in range(n):

pt = count

for col in range(max+1):

print(pt, end = " ")

pt -= 1

count = count + temp

temp += 1

print()

(05)

n = 4

Count = 1

for rows in range(n):

pt = count

for col in range(max+1):

print(pt, end = " ")

pt -= 1

Count = count + 2 + rows

print()

Output :-

1	2	1				
1	2	3	2	1		
1	2	3	4	3	2	1

\$

$$n = 4$$

for row in range(n):

Count = 1

for col in range(h-1-row):

print(" ", end = " ")

for star in range(2 \* row + 1):

print(" ", end = " ")

if row > star:

Count += 1

else:

Count -= 1

print()

logic

		1	2	3	2	1
		0=0	1=1	1<2		
2>0	2>1					
1	2	3	4	3	2	1
3>0	3>1	3>2	3=3	3<4	3<5	3<6
increment				decrement		
Count += 1				Count -= 1		

2) O/P :-

```
1 * 3  
1 * 3 * 5  
1 * 3 * 5 * 7
```

$\Rightarrow n = 4$

for row in range(n):

Count = 1

for space in range(n - 1 - row):

print(" ", end = "")

for star in range(2 \* row + 1):

if star % 2 == 0:

print(" ", end = "")

Count += 2

else:

print("\*", end = "")

print()

## Numerical Programming

Chandra's  
B.T. / Pg.

- 1) Extract the last digit in number  $\rightarrow$  num % 10
- 2) Eliminate the last digit in number  $\rightarrow$  num // 10

- 1) MAP to determine whether the given number is Armstrong number or not.  
**Armstrong number :-** Sum of power raised to length of given input of individual digit is equals to given number.

Steps :-

num = 153

- 1) Extract the last digit
  - 2) calculate the last digit power raised to total no. of digit and it to the output variable
  - 3) eliminate last digit.
  - 4) Repeat Step 1 to Step 3 until while
  - 5) compare the result of previous steps with given number and return the result.
- $\Rightarrow$  def nofDigits(num):

Count = 0

while num != 0:

    num = num // 10

    Count += 1

return Count

def power(num, power):

    out = 1

    for i in range(power):

        out \*= num

    return out

def isArmstrong(num):

    temp = num

    C1 = nofDigits(num)

    out = 0

```
while num != 0 :
    last = num % 10
    out = out + power( last, c1 )
    num = num // 10
return temp == out
```

num = 35

answer = isArmstrong(num)

if answer :

```
print( " Armstrong number" )
```

else :

```
print( " not an Armstrong number" )
```

2) WAP to display the Armstrong numbers from

1 to 100000



n = 1000

```
for i in range(1, n+1):
```

answer = isArmstrong(i)

if answer :

```
print(i)
```

3) WAP to display first 12 Armstrong numbers.

→ Count = 12

num = 1

while Count > 0 :

answer = isArmstrong(num)

if answer :

```
print(num)
```

Count = 1

num += 1

4) WAP to determine whether the given number is perfect number or not.

→ def perfect (num):

    Sum = 0

    for i in range(1, num//2 + 1):

        if num % i == 0 :

            Sum += i

    return Sum == num

num = 6

answer = ~~perfect~~ perfect(num)

if answer :

    print ("Perfect number")

else:

    print ("not a perfect number")

5) WAP to determine whether the given number is Strong number or not.

→ def fact (num):

    out = 1

    for i in range (num, 1, -1):

        out \*= i

    return out

num = 145

answer = fact (num)

if answer :

    print ("Strong number")

else:

    print ("not a Strong number")