

Easy Q1

SET A

Write a Python program to generate the following pattern below.

Note: This pattern is for the input value, N=5. There is no empty line between two printed lines.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Input format:

Line1: value of N

Output format:

Pattern

Sample Test case:

Input:

5

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Test case 1

Input:

10

Output:

```
1
1 2
1 2 3
```

```
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

Test case 2

Input:

7

Output:

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

Test case 3

Input:

3

Output:

```
1
1 2
1 2 3
```

```
rows=int(input())
spaces=rows
for r in range(1,rows+1):
    for c in range(spaces):
        print(end=' ')
    for c in range(r):
        print(c+1,end=' ')
    spaces-=1
    print()
```


SET B

Write a Python program to generate the following pattern below.

Note: This pattern is for the input value, N=5. There is no empty line between two printed lines.

```
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

Input format:

Line1: value of N

Output format:

Pattern

Sample Test case:

Input:

5

Output:

```
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

Test case 1

Input:

9

Output:

```
    1
   1 2
  1 2 3
```

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

Test case 2

Input:

7

Output:

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

Test case 3

Input:

3

Output:

```
      1
    1 2
  1 2 3
```

```
rows=int(input())
spaces=(rows*2)-2
for r in range(1,rows+1):
    for c in range(spaces):
        print(end=' ')
    for c in range(r):
        print(c+1,end=' ')
    spaces-=2
```

print()

SET C

Write a Python program to generate the following pattern below.

Note: This pattern is for the input value, N=5. There is no empty line between two printed lines.

```
5 4 3 2 1
 4 3 2 1
  3 2 1
   2 1
    1
```

Input format:

Line1: value of N

Output format:

Pattern

Sample Test case:

Input:

5

Output:

```
5 4 3 2 1
 4 3 2 1
  3 2 1
   2 1
    1
```

Test case 1

Input:

9

Output:

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

```
6 5 4 3 2 1
 5 4 3 2 1
  4 3 2 1
   3 2 1
    2 1
     1
```

Test case 2

Input:

7

Output:

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

Test case 3

Input:

3

Output:

```
3 2 1
 2 1
 1
```

```
rows=int(input())
spaces=0
for r in range(rows,0,-1):
    for c in range(spaces):
        print(end=' ')
    for c in range(r,0,-1):
```



```
print(c,end=' ')
spaces+=2
print()
```

Medium - Q2

SET A

Write a Python program to enter a positive integer N as input and print the sum of odd digits in the number.

e.g.,
if N is 8356; sum of odd digits = $3+5 = 8$

Input format:

Line1: value of N

Output format:

Line 1: sum of the odd digits in N

Sample Test case:

Input:
8356

Output:
8

Test case 1:

Input:
736593

Output:
27

Test case 2:

Input:
4682

Output:
0

Test case 3:

Input:
353

Output:
11

***num = n = int(input())
sumn = 0***

***while num>0:
 digit=num%10
 num=int(num/10)
 if digit%2!=0:
 sumn+=digit
print(sumn)***

SET B

Write a Python program to enter a number N and a digit D as inputs, and print the number of occurrences of D in N.

Input format:

Line1: Value of number, N

Line2: Value of digit, D

Output format:

Line 1: Number of times D occur in N

Sample Test case:

Input:

673852387463

3

Output:

3

Test case 1:

Input:

43242489

6

Output:

0

Test case 2:

Input:

554514353455

5

Output:

6

Test case 3:

Input:

888

8

Output:

3

Test case 4:

Input:

9

9

Output:

1

num = n = int(input())

digit=int(input())

count = 0

while num>0:

dig=num%10

num=int(num/10)

if dig==digit:

count+=1

print(count)

SET C

Write a Python program to enter a number N and a digit D as inputs, and print the number of digits that are less than D in N.

Input format:

Line1: Value of number, N

Line2: Value of digit, D

Output format:

Line 1: Number of digits less than D in N

Sample Test case:

Input:
536425
4

Output:
2

Test case 1:

Input:
98788
5

Output:
0

Test case 2:

Input:
324
4

Output:

2

Test case 3:

Input:

67548122

9

Output:

8

```
num = n = int(input())  
digit=int(input())  
count = 0
```

```
while num>0:  
    dig=num%10  
    num=int(num/10)  
    if dig<digit:  
        count+=1  
print(count)
```

Medium - Q3

SET A

You are transporting some boxes through a tunnel. A box can be carried through the tunnel only if its width is strictly less than a value W, and height is strictly less than a value H. Print the volume of each box that can be successfully transported to the other end of the tunnel. Note: Boxes cannot be rotated.

Input Format

Line 1: Values of W and H separated by space

Line 2: number of boxes, N.

Line 3: length width and height of box1, separated by space

Line 4: length width and height of box2, separated by space

.
.
.

Output Format

For each box that can be carried through the tunnel, print its length, width, height, and volume in a line(separated by space).

Sample Input

70 40

4

5 5 5

1 2 40

10 5 30

7 72 34

Sample Output

5 5 5 125

10 5 30 1500

Explanation:

The first box has weight 5 which is less than 70, and height 5 which is less than 40. Hence, the length, width, height and volume of the first box is printed out. Similarly for 3rd box.

Test cases:

Test case 1:

Input:

100 400

10

150 50 300
76 150 650
142 78 399
210 205 50
12 178 450
118 46 150
79 70 510
40 18 780
90 99 250
700 56 516

Output:

150 50 300 2250000
142 78 399 4419324
118 46 150 814200
90 99 250 227500

Test case 2:

Input:

560 45
5
79 560 34
40 18 780
190 99 250
700 56 45
76 250 650

Output:

NIL

Test case 3:

Input:

15 30
6
15 50 3
76 10 16
12 15 23
20 20 50
12 17 45
18 46 15

Output:

76 10 16 12160

```
W,H=input().split()
W=int(W)
H=int(H)
N=int(input())
count=0
for i in range(N):
    l,w,h = input().split()
    l=int(l)
    w=int(w)
    h=int(h)

    if w<W and h<H:
        print(l,w,h,l*w*h)
        count=1
if count==0:
    print('NIL')
```

SET B

Given a number N as input, write a Python program to generate the Hailstone sequence.

The next number in Hailstone sequence can be generated from the current number by following this rule:

- If the current number is even, divide it by 2;
- If the current number is odd, multiply it by 3 and add 1.
- If the current number is 1, end the sequence.

The sequence should start from the number N, and end at the value 1.

Input Format:

Line 1: Value of N

Output Format:

Line 1: Hailstone sequence, space separated list of numbers

Sample Test case:

Input:

10

Output (Hailstone Sequence):

10 5 16 8 4 2 1

Test cases:

Test case 1:

Input :

17

Output :

17 52 26 13 40 20 10 5 16 8 4 2 1

Test case 2:

Input :

7

Output :

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Test case 3:

Input :

1

Output :

1

```
n = int(input())
while n>1:
    print(n,end=' ')
    if n%2==0:
        next=int(n/2)
    else:
        next=3*n+1
    n=next
print(n)
```

SET C

Given a number N as input, write a Python program to generate a sequence according to the rules given below. Rules to generate the next number in the sequence:

- If the current number is odd, find the integer division of the current number by 3.
- If the current number is even, divide it by 2;

The sequence starts from the number N, and ends at the value 1.

Input Format:

Line 1: Value of N

Output Format:

Line 1: sequence (space separated list of numbers)

Sample Test case:**Input:**

10

Output:

10 5 1

Test cases:

Test case 1:

Input :

52

Output :

52 26 13 4 2 1

Test case 2:

Input :

97

Output :

97 32 16 8 4 2 1

Test case 3:

Input :

256

Output :

256 128 64 32 16 8 4 2 1

n = int(input())

while n>1:

print(n,end=' ')

if n%2==0:

next=int(n/2)

else:

next=int(n//3)

```
n=next  
print(n)
```

Hard Q1

SET A

Write a program to calculate and print the Electricity bill of a given customer.

The customer id, name, and number of units consumed by the user should be taken as inputs. Display the customer details and the total amount to be paid by the customer. The charges are as per the following slabs:

Unit	Charge/unit
1 - 150 units	50 paise
151- 400 units	Rs.1.20
401 - 670 units	Rs.1.80
671 and above	Rs.2.00

If the bill exceeds Rs.400, then an additional surcharge of 15% will be charged.

Input format:

Line1: customer id, name, and number of units consumed(space separated values)

Output format:

Line 1: Customer Id and name

Line 2: Number of units consumed and the total charge

Sample Test case:

Input:

1001 James 640

Output :

1001 James

640 928.05

Calculation:

charges for 640 units = $150 \times 50 \text{ paise} + 250 \times \text{Rs. } 1.20 + 240 \times \text{Rs. } 1.80 = 807$

Total charges = $807 + 15\% \text{ of } 807 = 928.05$

Note: Display only two decimal places of fractions.

Id,name,units=input().split()

units=int(units)

if units<=150:

charge=units*0.5

elif units>150 and units<=400:

charge=150*0.5 + (units-150)*1.20

elif units>400 and units<=670:

charge=150*0.5 + 250*1.20 + (units-400)*1.80

else:

charge=150*0.5 + 250*1.20 + 270*1.80 + (units-670)*2

if charge>400:

*charge=charge+(0.15*charge)*

print(Id,name)

print(units,'% .2f' %charge)

SET B

A library charges fine for every book that is returned late. For the first 5 days the fine is 50 paise per day, from 6th to 12th day, fine is Rs.1.50 per day and for above 12 days, fine is Rs.5 per day. If you return the book after 30 days, your membership will be cancelled. Write a program to accept the number of days the member is late to return the book and display the fine or the 'Membership cancelled' message.

Input format:

Line1: User id, name, and number of days late to return the book(space separated values)

Output format:

Line 1: User id, name, and number of days late to return the book

Line 2: Total fine amount

Calculation:

fine for 12 days = 5 * 50 paise + 7 * Rs. 1.50 = 13

Sample Test case:

Input:

5058 Luke 25

Output :

5058 Luke 25

Input:

6002 Peter 50

Output :

6002 Peter 50

Membership cancelled

SET C

A xerox centre charges the xerox copies taken by the customers using the following price chart. For the first 10 copies, charge is Rs. 3 per paper; from 11 to 20 copies, charge is Rs. 1.50 per paper; and for more than 20 copies, charge is 50 paise per paper. For the customers who take more than 100 copies, an additional discount of 10% is given on the total charge. Write a Python program to accept the number of xerox copies taken by a customer and print the total amount to be paid at the centre.

Input format:

Line1: customer name and number of xerox copies (space separated values)

Output format:

Line 1: customer name, number of xerox copies, and total amount

Line 2: If eligible for discount, output "Discount added"

Calculation:

charges for 15 copies = $10 * 3 + 5 * 1.5 = 37.50$

Sample Test case:

Input:

Dave 15

Output :

Dave 15 37.50

Input:

Luke 155

Output :

Luke 155 101.25

Discount added

```
name,copies=input().split()
```

```
copies=int(copies)
```

```
if copies<=10:
```

```
    charge=copies*3
```

```
    print(name,copies,charge)
```

```
elif 10<copies<=20:
```

```
    charge=10*3.0 + (copies-10)*1.50
```

```
    print(name,copies,'%.2f' %charge)
```


elif 20<copies<=100:

*charge=10*3.0 + 10*1.50 + (copies-20)*0.50*

print(cn,copies,format(t,'.2f'))

else:

*charge=10*3.0 + 10*1.50 + (copies-20)*0.50*

*charge=charge-(0.10*charge)*

print(name,copies,'% .2f' %charge)

print("Discount added")

Input:

SET C

Write a program asks the user to enter an amount and prints the minimum number of notes (of denominations 500, 100, 50, 20, 10, 1) to be distributed. For example, if the user enters Rs. 451, then 4 note of 100, 1 note of 50 and 1 note of 1 is required.

1. Make a two-player Rock-Paper-Scissors game. (*Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game*)

Remember the rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock

```
import sys
```

```
user1 = input("What's your name?")
```

```
user2 = input("And your name?")
```

```
user1_answer = input("%s, do yo want to choose rock, paper or scissors?" % user1)
```

```
user2_answer = input("%s, do you want to choose rock, paper or scissors?" % user2)
```

```
def compare(u1, u2):
```

```
    if u1 == u2:
```

```
        return("It's a tie!")
```

```
    elif u1 == 'rock':
```

```
        if u2 == 'scissors':
```

```
            return("Rock wins!")
```

```
        else:
```

```
            return("Paper wins!")
```

```
    elif u1 == 'scissors':
```

```
        if u2 == 'paper':
```

```
            return("Scissors win!")
```

```
        else:
```

```
            return("Rock wins!")
```

```
    elif u1 == 'paper':
```

```
        if u2 == 'rock':
```

```
            return("Paper wins!")
```

```
        else:
```

```
            return("Scissors win!")
```

```
    else:
```

```
        return("Invalid input! You have not entered rock, paper or scissors, try again.")
```

```
    sys.exit()
```

```
print(compare(user1_answer, user2_answer))
```

SET A

Write a Python program to generate the following pattern below.

Note: This pattern is for the input value, N=5.

5

5 4

5 4 3
5 4 3 2
5 4 3 2 1

Input format:

Line1: value of N

Output format:

Pattern

Sample Test case:

Input:

5

Output:

5
5 4
5 4 3
5 4 3 2
5 4 3 2 1

Test case 1

Input:

10

Output:

10
10 9
10 9 8
10 9 8 7
10 9 8 7 6
10 9 8 7 6 5
10 9 8 7 6 5 4
10 9 8 7 6 5 4 3
10 9 8 7 6 5 4 3 2
10 9 8 7 6 5 4 3 2 1

Test case 2

Input:

1

Output:

1

Test case 3

Input:

3

Output:

3

3 2

3 2 1

```
rows=int(input())  
for r in range(1,rows+1):  
    for c in range(r):  
        print(rows-c,end=' ')  
    print()
```

Write a Python program to construct the following pattern, using a nested for loop.

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

Write a Python program to print alphabet pattern 'Z'

SET C

Browsing Problem

Given the number of hours and minutes browsed, write a program to calculate the bill for Internet Browsing in a browsing center. The conditions are given below.

(a) 1 Hour Rs.50

(b) 1 minute Re. 1

(c) Rs. 200 for five hours

Boundary condition: User can only browse for a maximum of 7 hours

Check boundary conditions

Test Cases:

Input

Hours = 6

Minutes = 21

Output

Amount = 271

Processing Involved

Amount = 200 for first five hours

50 for sixth hour

21 for each minute

4. Write a Python program to calculate a dog's age in dog's years.

Note: For the first two years, a dog year is equal to 10.5 human years.

After that, each dog year equals 4 human years.

Expected Output:

Input a dog's age in human years: 15

The dog's age in dog's years is 73

5. Converting numbers from decimal to binary using while loop