One hot summer day Pete and his friend Billy decided to buy a watermelon. They chose the biggest and the ripest one, in their opinion. After that the watermelon was weighed, and the scales showed *w* kilos. They rushed home, dying of thirst, and decided to divide the berry, however they faced a hard problem.

Pete and Billy are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. The boys are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the watermelon in the way they want. For sure, each of them should get a part of positive weight.

**Input**

The first (and the only) input line contains integer number *w* (1 ≤ *w* ≤ 100) — the weight of the watermelon bought by the boys.

**Output**

Print YES, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and NO in the opposite case.

**Examples**

**input**

**Copy**

8

**output**

**Copy**

YES

CODE

def can\_divide\_watermelon\_evenly(w):

if w > 2 and w % 2 == 0:

return "YES"

else:

return "NO"

# Example usage:

w = int(input())

print(can\_divide\_watermelon\_evenly(w))

An elephant decided to visit his friend. It turned out that the elephant's house is located at point 0 and his friend's house is located at point *x*(*x* > 0) of the coordinate line. In one step the elephant can move 1, 2, 3, 4 or 5 positions forward. Determine, what is the minimum number of steps he need to make in order to get to his friend's house.

**Input**

The first line of the input contains an integer *x* (1 ≤ *x* ≤ 1 000 000) — The coordinate of the friend's house.

**Output**

Print the minimum number of steps that elephant needs to make to get from point 0 to point *x*.

**Examples**

**input**

**Copy**

5

**output**

**Copy**

1

**input**

**Copy**

12

**output**

**Copy**

3

**Note**

In the first sample the elephant needs to make one step of length 5 to reach the point *x*.

In the second sample the elephant can get to point *x* if he moves by 3, 5 and 4. There are other ways to get the optimal answer but the elephant cannot reach *x* in less than three moves.

CODE:

distance = int(input())

steps\_with\_5 = distance//5

remain = distance%5

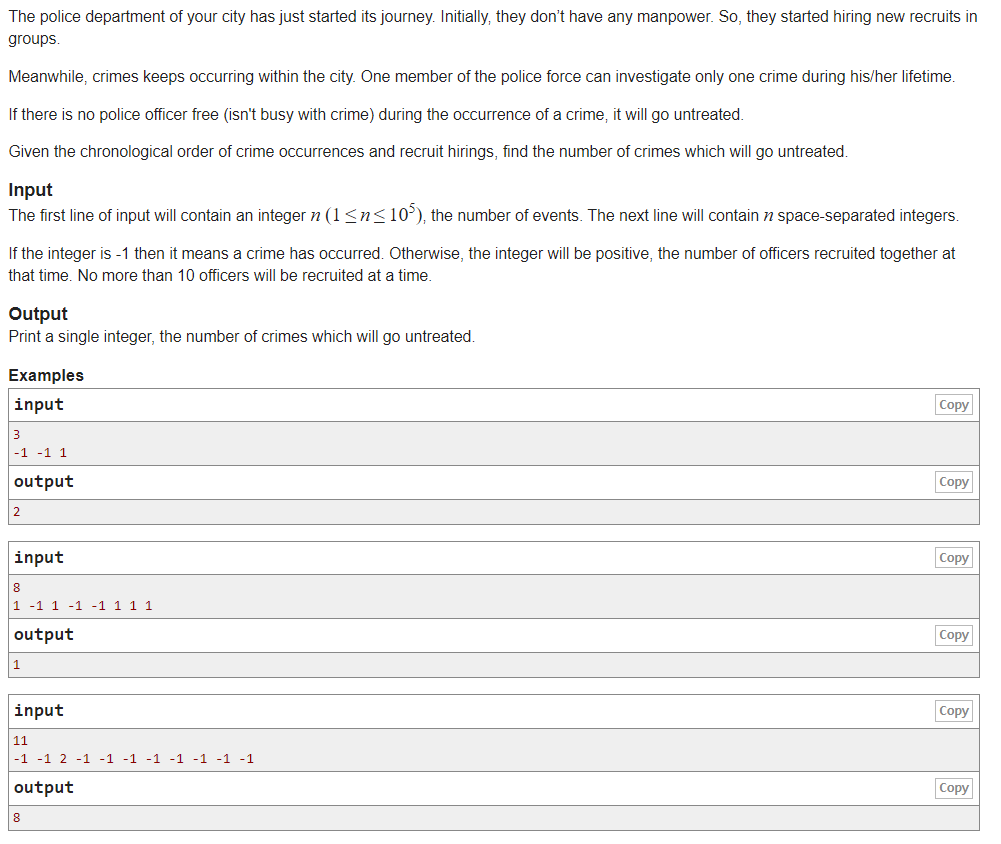
if remain > 0:

    steps = steps\_with\_5 + 1

else:

    steps = steps\_with\_5

print(steps)



CODE:

ev = int(input())

l = list(map(int,input().split()))

for i in l:

    av  = 0

    ut = 0

    if i == -1:

        if av > 0:

            av -= av

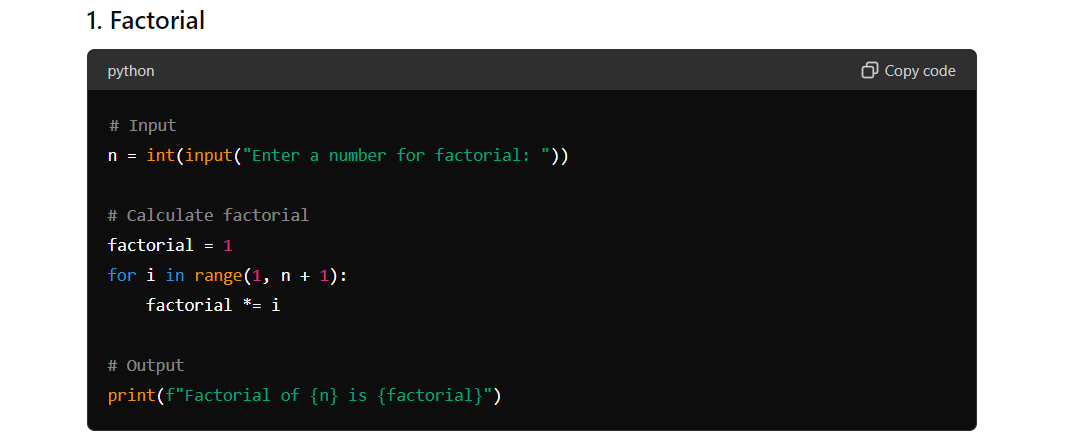
        else:

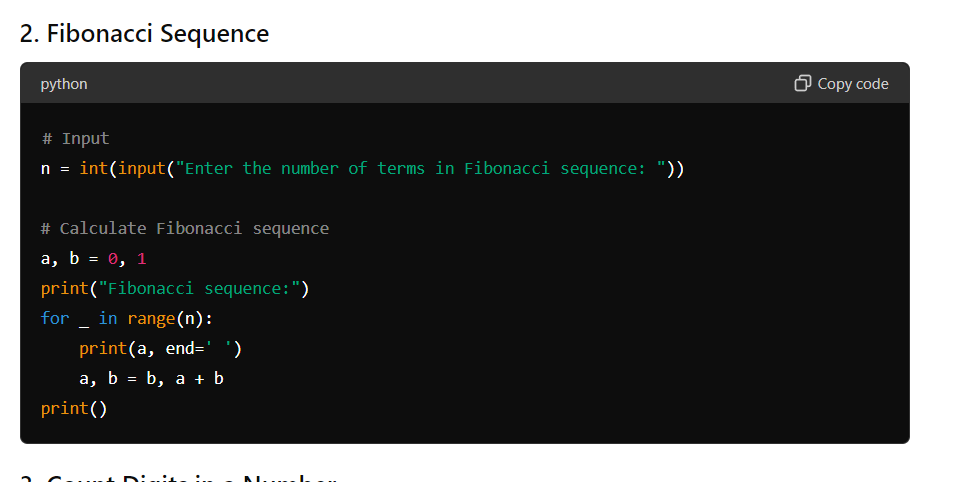
            ut += 1

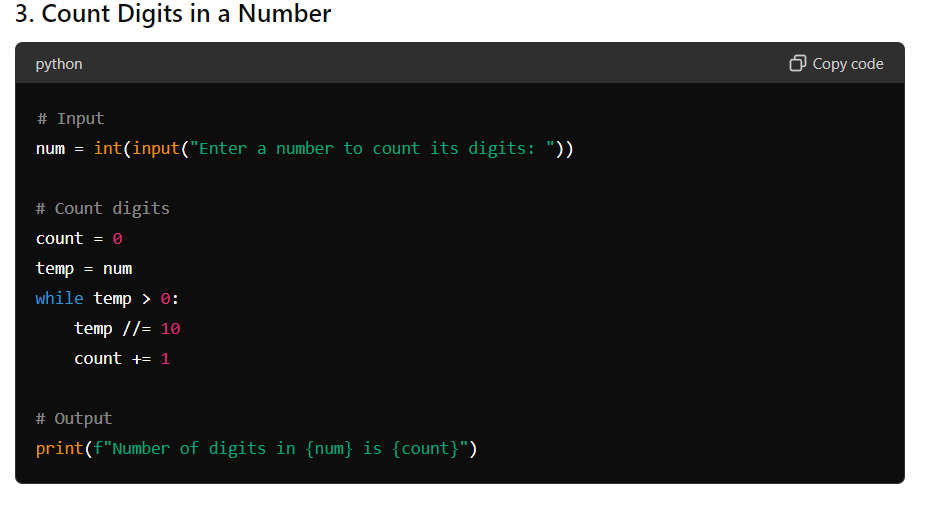
    else:

        av += 1

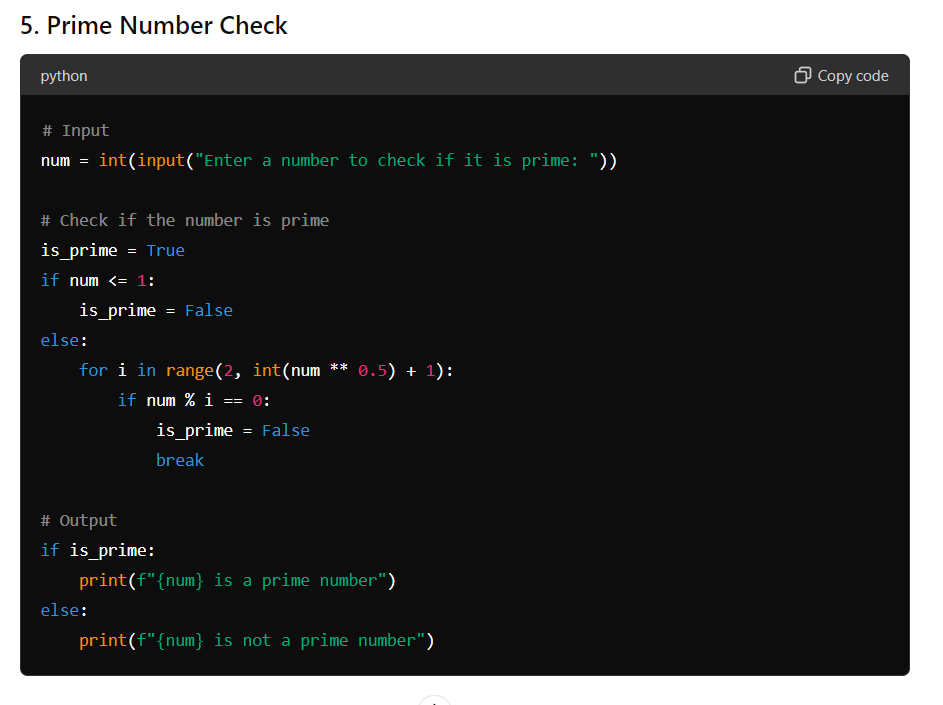
print(ut)











Bear Limak wants to become the largest of bears, or at least to become larger than his brother Bob.

Right now, Limak and Bob weigh *a* and *b* respectively. It's guaranteed that Limak's weight is smaller than or equal to his brother's weight.

Limak eats a lot and his weight is tripled after every year, while Bob's weight is doubled after every year.

After how many full years will Limak become strictly larger (strictly heavier) than Bob?

**Input**

The only line of the input contains two integers *a* and *b* (1 ≤ *a* ≤ *b* ≤ 10) — the weight of Limak and the weight of Bob respectively.

**Output**

Print one integer, denoting the integer number of years after which Limak will become strictly larger than Bob.

**Examples**

**input**

**Copy**

4 7

**output**

**Copy**

2

**input**

**Copy**

4 9

**output**

**Copy**

3

**input**

**Copy**

1 1

**output**

**Copy**

1

**Note**

In the first sample, Limak weighs 4 and Bob weighs 7 initially. After one year their weights are 4·3 = 12 and 7·2 = 14 respectively (one weight is tripled while the other one is doubled). Limak isn't larger than Bob yet. After the second year weights are 36 and 28, so the first weight is greater than the second one. Limak became larger than Bob after two years so you should print 2.

In the second sample, Limak's and Bob's weights in next years are: 12 and 18, then 36 and 36, and finally 108 and 72 (after three years). The answer is 3. Remember that Limak wants to be larger than Bob and he won't be satisfied with equal weights.

In the third sample, Limak becomes larger than Bob after the first year. Their weights will be 3 and 2 then.

CODE:

a,b = 4,7

count = 0

while a <= b:

    a = a\*3

    b = b\*2

    count += 1

print(count)

**DAY – 03**

**XoR:**

Following a Pattern

Accordingly,

N%4 == 1, OP = 1

N%4 ==2, OP = N +1

N%4 == 3, OP = 0

N%4 == 0, OP = N

n = int(input())

"""a = 0

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

    a ^= i

print(a)"""#### for xor between certain numbers

if (n % 4 == 1):

    print(1)

if (n % 4 == 2):

    print(n+1)

if (n % 4 == 3):

    print(0)

if (n % 4 == 0):

    print(n)

1 – 1 -> 1

2 – 3 ->N+1

3 – 0 -> 0

4 – 4 -> N

5 – 1 -> 1

6 – 7 ->N+1

7 – 0 -> 0

8 – 4 -> N

**Write a program for Password check:**

correct\_password = "hello\_namste"

attempts = 0

max\_attempts = 3

while attempts < max\_attempts:

inputp = input("Enter your password: ")

if inputp == correct\_password:

print("Correct password")

break

else:

attempts += 1

print("Incorrect password. You have", max\_attempts - attempts, "attempt(s) left.")

# If the user fails to enter the correct password within the maximum attempts

if attempts == max\_attempts:

 print("Locked")

**input a list and sort the elements in the list based on the length**

code:

a = input().split()

a.sort(key = len)

print(a)

**SWAP**

a = input().split()

temp = a[-1]

a[-1] = a[0]

a[0] = temp

print(a)

print the elements in list which occurred odd number of times

a = input().split()

b = []

for i in a:

    if a.count(i)%2 != 0 and i not in b:

        b.append(i)

print(b)

**Take a nested list and sort the list according to the second element in the list**