**Q.1**

n = input()#7564168

otp =''

for i in range(1,len(n),2):

otp += str(int(n[i])\*\*2)

print(otp[:4])

**Q.2**

s = input()

n = s.split(',')

result = ''

for i in n:

nc = i.split(':')

name = nc[0]

code = nc[1]

max = 0

for i in code:

if i.isdigit() and int(i) > max and int(i) <= len(name):

max = int(i)

if max > 0:

result += name[max - 1]

else:

result += 'X'

print(result)

**Q.3**

def find\_unbalanced\_position(expression):

stack = []

pairs = {")": "(", "]": "[", "}": "{"}

for i, symbol in enumerate(expression):

if symbol in "([{":

stack.append((symbol, i))

elif symbol in ")]}":

if not stack:

return i

top\_symbol, \_ = stack.pop()

if top\_symbol != pairs[symbol]:

return i

if stack:

\_, index = stack.pop()

return index

return 0

expression = input("Enter an expression: ")

position = find\_unbalanced\_position(expression)

if position == 0:

print("The expression is balanced.",position)

else:

print(f"The expression is not balanced. Error at index {position+1}.")

**Q.4:**

def demo(s):

l=[]

for i in s:

if i.isalpha():

l.append(i)

else:

spc = i

idxspc = s.index(i)

l.reverse()

l.insert(idxspc,spc)

return ''.join(l)

s = input()

print(demo(s))

**Q.5:**

def find\_pairs\_with\_sum(n, target):

pairs = []

for i in range(len(n)):

for j in range(i, len(n)):

if n[i] + n[j] == target:

pairs.append((i, j))

return pairs

if \_\_name\_\_ == '\_\_main\_\_':

n = list(map(int, input("Enter a list of numbers: ").split(',')))

target = int(input("Enter the target sum: "))

pairs = find\_pairs\_with\_sum(n, target)

if pairs:

print("Pairs with sum", target, "are at index positions:")

for pair in pairs:

print(pair)

else:

print("No pairs found with sum", target)

**Q.6**

def count\_numbers\_at\_least\_k\_times(n, k):

num\_count = {}

for num in n:

if num in num\_count:

num\_count[num] += 1

else:

num\_count[num] = 1

count = 0

for num, freq in num\_count.items():

if freq >= k:

count += 1

return count

if \_\_name\_\_ == '\_\_main\_\_':

n, k = map(int, input().split())

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

result = count\_numbers\_at\_least\_k\_times(arr, k)

print(result)