class solution:

def lenghtoflastword(self, s: str)-> int:

s=s.strip()

if s=="":

return 0

elif len(s.split())==1:

return len(s)

return len(s.split()[-1])

ob=solution()

print(ob.lenghtoflastword("fly me tothe moon"))

grade=input("enter the grade of employee")

salary=int(input("enter the salary of the employee"))

if grade=='A':

if salary < 10000:

bonus=salary\*0.07

else:

bonus=salary\*0.05

print(bonus)

salary = salary + bonus

print(salary)

elif grade =='B':

if salary < 10000:

bonus=salary\*0.12

else:

bonus = salary\*0.1

print("bonus:",bonus)

salary = salary + bonus

print("total to bepaid:",salary)

else:

print(salary)

def getMinSquare(n):

if n <= 3:

return n;

res = n

for x in range(1,n +1..):

temp = x\*x;

if temp > n:

break

else:

res = min(res, 1 + getMinsquare(n - temp))

return res;

print(getMinSquare(4))

data = input()

li = data.split()

a= int(li[0])

b= int(li[1])

def gcd(a,b):

if (a == 0):

return b;

rerturn gcd(b%a,a);

if(a > 0 and a < (10\*\*12+1) and b >=1 and b < (10\*\*12+1)):

count = 1

for i in range(2, gcd(a,b)+1):

if a%i == 0 and b % i == 0:

count = count +1

print(count)

def findpeak(num, left=none, righr=none):

if left is none and right is none:

left, right = 0,len(nums)-1

mid = (left+right)//2

if ((mid == 0 or nums[mid - 1] <=nums[mid])and (mid == len(nums) - 1 or nums[mid+1] <= nums[mid])):

return mid

if mid - 1 >= 0 and nums[mid -1] > nums[mid]:

return findpeak(num,left,mid-1)

return findpeak(nums,mid +1,right)

def findpeakElement(nums) -> int:

if not nums:

exit(-1)

index = findpeak(nums)

return nums[index]

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

nums = [5,10,20,15]

print('the peak element is,'findpeakElement(nums))

from math import factorial

n = 5

for i in range(n):

for j in range(n-i+1):

print(end="")

for j in range(i+1):

print(factorial(i)//(factorila(j)\*factorial(i-j)),end="")

print()

MAX = 26

def atleastk(freq,k):

for i in range(MAX):

if(freq[i] !=0 and freq[i] < k):

return false;

return ture;

def setzero(freq):

for i in range(MAX):

freq[i]= 0;

def findlenght(string,n,k):

maxlen =0;

fre[i] = 0;

def findlenght(string,n,k):

maxlen = 0;

freq = [0]\*MAX;

for in range(n):

setZero(freq);

for j in range(i,n):

freq[ord(string[j])- ord('a')] += 1;

if (atleast(freq,k)):

maxlen = max(maxlen,j -i + 1);

return maxLen;

if\_name\_ == "\_main\_":

string = "aaabb";

n =len(string);

k = 3;

print(findlenght(string, n, k));

def movesTochessboard(board):

n = len(board)

for r in range(0,n):

for c in range(0,n):

if(board[0][0] == 1):

return -1

rowsum = 0

colsum = 0

rowswap = 0

colswap= 0

for i in range(o,n):

rowsum += board[i][0]

colsum += board[0][i]

rowswap +=board[i][0] == i % 2

colswap +=board[0][i] == i % 2

if (rowsum != n//2 and rowsum != (n+1)//2):

return -1

if (colsum != n//2 and colsum !=(n+1)//2):

return -1

if (n%2):

if (rowswp%2):

rowswap = n - rowswap

if (colswap%2):

colswap = n - colswap

else:

rowswap = min(rowswap, n- rowswap)

colswap = min(colswap, n - colswap)

return(rowswap + colwap)//2

if\_name\_ == "\_main\_":

arr = [[0,1,1,0]],[0,1,1,0],[1,0,0,1],[1,0,0,1]]

minswap = movestochessboard(arr)

if (minswap == -1):

print("impossible")

else:

print(minswap)

def shuffle(11,12):

c=[]

if len(11)!=0 and len(12)!=0:

for i in range(min(len(11),len(12))):

c.extend([11[i],12[i]])

c.extend(11[i+1:] or 12[i+1:])

else:

c.extend(11[0:] or 12[0:])

return (c)

print(shuffle([1,2,3,8],[9,14,5,0,7,3]))

string = "be good and do good"

s = string.split()[::-1]

1 = []

for i in s:

1.append(i)

print("".join(1))