3

1、

n = int(input())

print("\*"\*n)

2、

#0是48，A是65，大写比小写少32

#chr(数字) 数字转字符

#ord(字符) 字符转数字

n = input()

print(chr(ord(n)-32))

3、

#思路：枚举法:多数据下，找到满足条件的情况

n = int(input())

m = list(map(int,input().split(",")))

x = 0

for a in m:

for b in m:

if a+b == n and a != b:

x += 1

print(x/2)

\*4、思路：大于平均值的小朋友要分给旁边的人，所以对小朋友们的糖果进行分析，先统一向右传，再统一向左传，通过判断糖果数量是否大于平均数后，再去判断之后的总和是否右剩余，有剩余就向另一侧传，少于平均值就想此侧传递。

lst = list(map(int,input().split(",")))

n = 0 #传递次数

ping = sum(lst)/len(lst)

max\_n = max(lst)

#从左到右

for i in range(len(lst)):

if lst[i] > ping and i <= len(lst)-2:

n += 1

x = (len(lst)-i-1)\*ping - sum(lst[i+1:])

if x >= (lst[i]-ping):

lst[i+1] += lst[i]-ping

lst[i] = ping

else:

lst[i+1] += x

lst[i] -= x

#从右到左

for i in range(len(lst)):

if lst[len(lst)-i-1] > ping and i <= len(lst)-2:

n += 1

x = (len(lst)-i-1)\*ping - sum(lst[:len(lst)-i-1])

if x >= (lst[len(lst)-i-1]-ping):

lst[len(lst)-i-2] += lst[len(lst)-i-1]-ping

lst[len(lst)-i-1] = ping

else:

lst[len(lst)-i-2] += x

lst[len(lst)-i-1] -= x

print(n)

\*5、

A=[]

def abc(list1,left,rihgt):##判断在交换过程中是否出现'1\*003\*04'

global A

r = rihgt

for i in range(rihgt - 1, left, -1):##这里一定倒数循环，正面循环会提前结束

if list1[left+1] == '0'and list1[left+2]!='\*':

list1.pop(left+1)##如果存在第一个 \* 后面 0 话 删除的之后，left的位置是不变的，继续接着判断是否 0

r-=1

else:

break

for i in range(len(list1)-1,r+1,-1):

if list1[r+1]=='0':

list1.pop(r+1)

else:

break

A.append("".join(list1))

n=int(input())

a=list(input())

for i in range(1,len(a)-n+1):

b=a[:]

c=n

j = i

while c != 0:

b.insert(j, "\*")

j += 2

c -= 1

c=b[:]

if c[0]=='0'and c[1]!='\*':##'01\*23\*45'这种情况，一个剔除“0”

b.remove(c[0])

j -= 1

abc(b, j - 4, j - 2)##判断是否有这个类型的母类01\*0\*010

else:

abc(c, j - 4, j-2)

for z in range(j - 2, len(b) - 2):

##这里j-2原因是：j在while循环结束时，增加2，不是第二个"\*"的位置

##以一个基础类型1\*2\*3456，再通过循环交换方式，得到他的子类：1\*23\*456，1\*234\*56，1\*2345\*6

b[z],b[z+1]=b[z+1],b[z]

c=b[:]##重新将b数组数据导入到c中

abc(c,j-4,z+1)

print(A)

a.clear()

for i in range(len(A)):

a.append(eval(A[i]))

print(max(a))