# 231A. Teams, brute force/greedy

# <https://codeforces.com/problemset/problem/158/A>

n = 0  
for i in range(int(input())):  
 s = input()  
 n = (int(s[0]) + int(s[2]) + int(s[4]) >= 2) + n  
print(n)

# 1A. Theatre Square, math

# <https://codeforces.com/problemset/problem/1/A>

def qz(ka):  
 if ka == int(ka):  
 return ka  
 else:  
 return int(ka + 1)

这题报错了好几次，主要是对List

下标越界和一些边界上的特殊情况考虑不周

s = input()  
i = 0  
while s[i] != " ":  
 i = i + 1  
l = int(s[:i])  
s = s[i + 1:]  
i = 0  
while s[i] != " ":  
 i = i + 1  
print(round(qz(l / int(s[i + 1:])) \* qz(int(s[:i]) / int(s[i + 1:]))))

# 158A. Next Round, implementation

# <https://codeforces.com/problemset/problem/158/A>

s = input().split()  
i = int(s[1])  
k = int(s[0])  
lst = input().split()  
if lst[i - 1] != "0":  
 while i < k and lst[i] != "0" and int(lst[i]) == int(lst[i - 1]):  
 i = i + 1  
else:  
 while i > 0 and lst[i - 1] == "0":  
 i = i - 1  
print(i)

这题让我头疼了一晚上（>-<）

以为程序中能用0.99或0.999999的办法向上取整来应对大多数输入情况，结果因为算法不严谨和数据精度的问题吃了大亏。

教训：测试数据充分考虑了各种极端情况，别想着蒙混过关

# 50A. Domino pilling, greedy/math

# <https://codeforces.com/problemset/problem/50/A>

s = input().split()  
print(int(s[0]) \* int(s[1]) // 2)