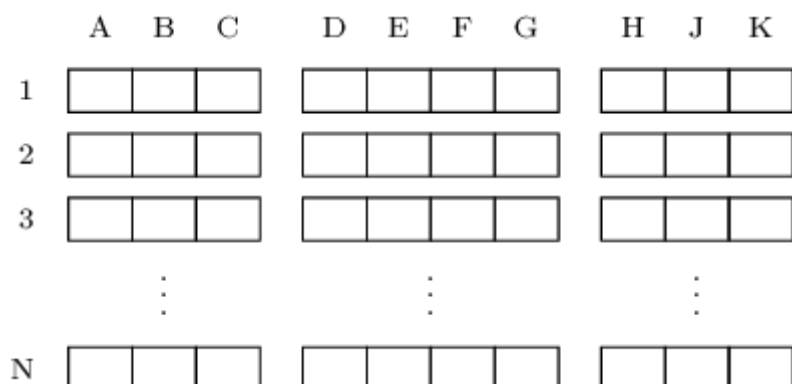


You are processing plane seat reservations. The plane has N rows of seats, numbered from 1 to N . There are ten seats in each row (labelled from A to K, with letter I omitted), as shown in the figure below:



Some of the seats are already reserved. The list of reserved seats is given as a string S (of length M) containing seat numbers separated by single spaces: for example, "1A 3C 2B 40G 5A". The reserved seats can be listed in S in any order.

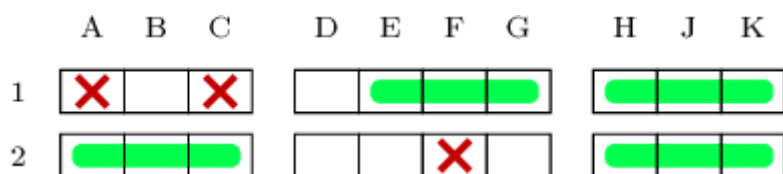
Your task is to allocate seats for as many three-person families as possible. A three-person family occupies three seats that are next to each other in the same row. Seats across aisle (such as 2C and 2D) are not considered to be next to each other. Obviously, each available seat cannot be taken by more than one family.

Write a function:

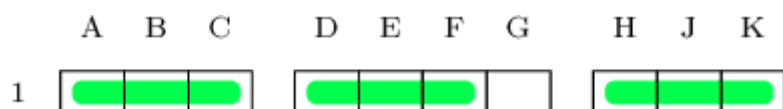
```
def solution(N, S)
```

that, given the number of rows N and a list of reserved seats as string S , returns the maximum number of three-person families that can be seated in the remaining unreserved seats.

For instance, given $N = 2$ and $S = "1A 2F 1C"$, your function should return 4. The following figure shows one possible way of seating four families in the remaining seats:



Given $N = 1$ and $S = ""$ (empty string), your function should return 3, because we can seat at most three families in a single row of seats, as shown in the figure below:



Assume that:

- N is an integer within the range [1..50];
- M is an integer within the range [0..1,909];
- string S consists of valid seat names separated with spaces;
- every seat number appears in string S at most once.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.