# Competitive Programming and Contests

## Xmas lights

In a famous street, there are $n$ houses. Each of these houses has Christmas lights of the owner's favorite color. There are three possible colors: **Red**, **White**, and **Green**. The colors are listed in an array $H[1, n]$, where $H[i]$ is the color of the $i$th house of the street.

You would like to compute the number of patriotic selections of three houses ($1 \leq i < j < p \leq n$) such that $H[i] =$ **Red**, $H[j] =$ **White** and $H[p] =$ **Green**. Note that houses are NOT required to be consecutive!

Once you realize that this computation can be easily done in linear time with a left-to-right scan of $H$ (How?), you want a much more difficult problem. What if $k$ houses have an unassigned color, say `X`, that can be set to any of the three colors above? Fixing these $k$ colors gives you one of $3^k$ possible different street colorings (and no, $k$ is not a constant as it could be up to $n$).

Dynamic programming helps you in computing the number of patriotic selections in those $3^k$ possible street colorings in linear time. Of course, it's not a great idea to list all the possible colorings.

**Input.** The first line contains $n$, the number of houses. The next line contains a character in { `R`, `W`, `G`, `X` } for each color of the $n$ houses..

**Output.** On a single line print the number of patriotic selections in the $3^k$ possible street colorings.

## Example

| Input | Output |
| --- | --- |
| 5<br>RWGXX | 16 |

Explaination:
The possible street colorings are
`R W G R R` with 1 patriotic selection
`R W G R W` with 1 patriotic selection
`R W G R G` with 2 patriotic selections
`R W G W R` with 1 patriotic selection
`R W G W W` with 1 patriotic selection
`R W G W G` with 3 patriotic selections
`R W G G R` with 2 patriotic selections
`R W G G W` with 2 patriotic selections
`R W G G G` with 3 patriotic selections