Passing time, walking the passage, as you pass the String Parsing Department(abbreviated SPaDe), you pause, amazed at them by parsing strings way past midnight. At the SPaDe , they are overwhelmed with the stringent requirements to compression recently introduced by the SPaDe's director, Dr. Spade. Any string longer than 4 characters must now be compressed as much as possible, Dr. Spade dictates! "If a string can be expressed shorter, so it must be!", he shouts.

He then yells that **abababab** can be expressed as just **(ab)4** which is only 5 symbols, a whole saving of 3 symbols, and everyone in the Department breaks out in a song of celebration, chanting:

```
This is why I'm hot
This is why I'm hot
This is why
This is why
This is why I'm hot
This is why I'm hot
This is why I'm hot
This is why
This is why
This is why I'm hot
```

but of course given in its compressed form

```
(This is why I'm hot)2
(This is why)2
(This is why I'm hot)3
(This is why)3
I'm hot
```

Given a string S over the alphabet {a,**b**,c,d} as input, output the length of its most compressed version. The SPaDe has yet to discover nested compression as in **((a)2b)3** so use only one-level compression

Input

The first line contains an integer T $(1 \leq T \leq 100)$, the number of test cases. For each test case there is a line with a string S $(5 \leq |S| \leq 100)$.

Output

For each test case, print on a separate line the minimum length of S after the compression described above.

Note: The string from the second example can be compressed into **d(abba)2ba(d)5cccc(b)12**.

Sample Input

```
2
abcda
dabbaabbabadddddccccbbbbbbbbbbbb
```

Sample Output

```
5
23
```