Problem Statement **Suggest Edit**

Given a string ‘pattern’, which contains only two types of characters ‘(’, ‘)’.

Your task is to find the minimum number of parentheses either ‘(’, ‘)’ we must add the parentheses in string ‘pattern’ and the resulted string is valid.

Condition for valid string-

Every opening parenthesis ‘(’ must have a correct closing parenthesis ‘)’.

Example - ‘(()(()))’, ‘()()()’, ‘((()))’ are valid string, and ‘(((’, ‘(()’, ‘)(())’ are invalid string.

Note:

```
1. You are not required to print the output explicitly, it has already been taken
care of. Just implement the function and return the minimum number of parentheses
required to make a string valid.
```

Input Format:

```
The first line of input contains an integer ‘T’ denoting the number of test cases.
The next ‘T’ lines represent the ‘T’ test cases.

The first line of each test case contains a string ‘pattern’.
```

Output Format:

```
For each test case, return the minimum number of parentheses.
```

Constraints:

```
1 <= T <= 50
1 <= N <= 10^4

Where ‘T’ is the total number of test cases, ‘N’ denotes the length of string
‘pattern’.

Time limit: 1 second
```

Sample Input 1:

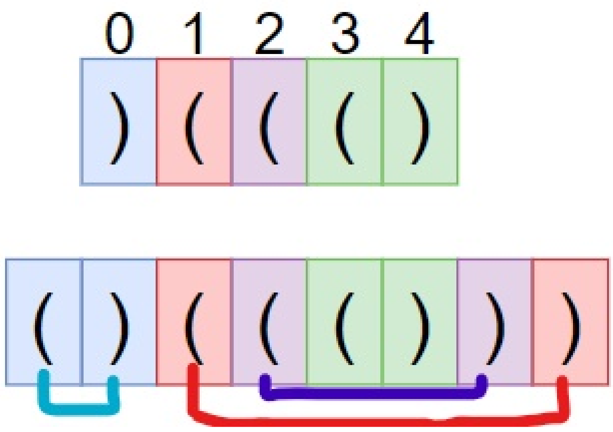
```
2
)((()
((
```

Sample Output 1:

```
3
2
```

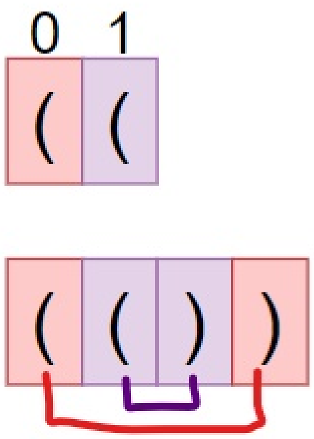
Explanation Of Sample Input 1:

```
Test Case 1:
```



```
In the given ‘pattern = )((()’, only one pair of parentheses ( at index ‘3’ and
‘4’) is valid and parentheses at index ‘0’, ‘1’ and ‘2’ are invalid.
As you can see, we need three extra parentheses to make the string valid in the
above image, one opening parenthesis ‘(’ for index ‘0’, two closing parentheses
‘)’ for index ‘1’ and ‘2’.
So return 3 number of minimum extra parentheses to make string ‘pattern’ valid.

Test Case 2:
```



```
In the given ‘pattern = ((,’ so we need two extra closing parentheses ‘)’ to make
string ‘pattern’ valid.
So return 2 number of minimum extra parentheses to make string ‘pattern’ valid.
```

Sample Input 2:

```
2
(((((
))(
```

Sample Output 2:

```
5
3
```