

括号序列 (bracket)

【题目描述】

小 w 在赛场上遇到了这样一个题：一个长度为 n 且符合规范的括号序列，其有些位置已经确定了，有些位置尚未确定，求这样的括号序列一共有多少个。

身经百战的小 w 当然一眼就秒了这题，不仅如此，他还觉得一场正式比赛出这么简单的模板题也太小儿科了，于是他把这题进行了加强之后顺手扔给了小 c。

具体而言，小 w 定义“超级括号序列”是由字符 (、)、* 组成的字符串，并且对于某个给定的常数 k ，给出了“符合规范的超级括号序列”的定义如下：

1、()、(S) 均是符合规范的超级括号序列，其中 S 表示任意一个仅由不超过 k 个字符 * 组成的非空字符串（以下两条规则中的 S 均为此含义）；

2、如果字符串 A 和 B 均为符合规范的超级括号序列，那么字符串 AB、ASB 均为符合规范的超级括号序列，其中 AB 表示把字符串 A 和字符串 B 拼接在一起形成的字符串；

3、如果字符串 A 为符合规范的超级括号序列，那么字符串 (A)、(SA)、(AS) 均为符合规范的超级括号序列。

4、所有符合规范的超级括号序列均可通过上述 3 条规则得到。

例如，若 $k = 3$ ，则字符串 ((**())*(*)*) (***) 是符合规范的超级括号序列，但字符串 *()、(*())、((**))、(****(*)) 均不是。特别地，空字符串也不被视为符合规范的超级括号序列。

现在给出一个长度为 n 的超级括号序列，其中有一些位置的字符已经确定，另外一些位置的字符尚未确定（用 ? 表示）。小 w 希望能计算出：有多少种将所有尚未确定的字符一一确定的方法，使得得到的字符串是一个符合规范的超级括号序列？

可怜的小 c 并不会做这道题，于是只好请求你来帮忙。

【输入格式】

从文件 *bracket.in* 中读入数据。

第 1 行，2 个正整数 n, k 。

第 2 行，一个长度为 n 且仅由 (、)、*、? 构成的字符串 S 。

【输出格式】

输出到文件 *bracket.out* 中。

输出一个非负整数表示答案对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```
1 7 3
2 (*??*??
```

【样例 1 输出】

```
1 5
```

【样例 1 解释】

如下几种方案是符合规范的：

```
1 (**) * ( )
2 (** ( * ))
3 (* ( ** ))
4 (*) ** ( )
5 (*) ( ** )
```

【样例 2 输入】

```
1 10 2
2 ??? (*??(??
```

【样例 2 输出】

```
1 19
```

【样例 3】

见选手目录下的 *bracket/bracket3.in* 与 *bracket/bracket3.ans*。

【样例 4】

见选手目录下的 *bracket/bracket4.in* 与 *bracket/bracket4.ans*。

【数据范围】

测试点编号	$n \leq$	特殊性质
1 ~ 3	15	无
4 ~ 8	40	
9 ~ 13	100	
14 ~ 15	500	S 串中仅含有字符?
16 ~ 20		无

对于 100% 的数据， $1 \leq k \leq n \leq 500$ 。