最大乘积

题目描述

2000年是国际数字联盟确定的“2000——世界数字年”，又恰逢我国著名数学家华罗庚先生诞辰90周年。在华罗庚先生的家乡江苏金坛，组织了一场别开生面的数学智力竞赛的活动，你的一个好　朋友XZ也有幸得以参加。活动中，主持人给所有参加活动的选手出了这样一道题目：

设有一个长度为N的数字串，要求选手使用K个乘号将它分成K+1个部分，找出一种分法，使得这K+1个部分的乘积最大。

同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：

有一个数字串：312，　当N=3，K=1时会有以下两种分法：

1）3*12=36

2）31*2=62

这时，符合题目要求的结果是：31*2=62。

现在，请你帮助你的好朋友XZ设计一个程序，求得正确的答案。

输入

第一行共有2个自然数N，K（6≤N≤10，1≤K≤6）
第二行是一个长度为N的数字串。

输出

输入样例

4 2
1231

输出样例

62

解析步骤

1、首先使用测试数据1231建立动态规划表Dp，横轴代表数字的个数，即前多少个数，我们用i来表示；纵轴代表乘号的数量，我们用j来表示。
那么该表合起来的含义就是前i个数中含有j个乘号时的值是多少。
很明显，该表中的第一行的含义为在前i个数字中，含有0个乘号的数值。

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1				
2				
3				

2、开始填表。要想数字中含有乘号，那么数字的数位至少要比乘号的数量多1，因此“在前1个数字中含有1个乘号的值是多少”，这样的描述是无意义的。所以，我们从数位2开始填充含有1个乘号的状况。
很明显，在前2个数中存在1个乘号的时候，这个值是2。

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2		
2				
3				

3、继续填表。我们开始求在前3个数字中含有1个乘号时的最大值。这个时候存在两种可能，我们分别计算。
a、第一种情况，乘号在1后，结果为23。
b、第二种情况，乘号在2后，结果为36。
C、所以，最终的最大值为36。

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	1x23=23	
2				
3				

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	12x3=36	
2				
3				

4、接下来，我们要计算在前4个数字中含有1个乘号时的最大值，这时候很明显存在3种情况。
a、1x231=231
b、12x31=372
c、123x1=123
因此，最大值为372

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	23	372
2				
3				

5、现在，我们开始填充第二行，即在前i个数字中含有2个乘号时的最大值，由于数字的位数必须大于2，所以，我们从3开始。
前3位数含有两个乘号时，结果只能为1x2x3=6

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	23	372
2			6	
3				

6、继续填充，在前4个数字中填充两个乘号，这时我们该如何选择呢？
答案是：这个时候我们要先选择第一个乘号出现的位置，一共有两种情况。

A、当第一个乘号出现在前两个数字之间时，这个最大值为Dp[2][1]=2,即左图中的蓝色值，这个时候再让2和我们当前剩下的31做乘法，结果为62。

B、当第一个乘号出现在前三个数字之间时，这个最大值为DP[3][1]=23，即右图中的蓝色值，同样的我们用23与剩下的1做乘法，结果为23。

所以，在前4个数中含有2个乘号时，最大值为62。

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	23	372
2			6	31x2=62
3				

乘号数量j	数位i			
	1	2	3	4
0	1	12	123	1231
1		2	23	372
2			6	23x1=23
3				

7、其实，看到这里我们已经能够推出我们的状态转移方程了，它就是：
Dp[前i个数][j个乘号]=max(Dp[前i个数][j个乘号],Dp[前j+1至i-1个数][j-1个乘号]*剩余数值)

因为j-1个乘号可能出现在多个位置，所以我们必须不断的遍历尝试，这个尝试的范围就是“**前j+1至i-1个数**”，在上面的“前4个数中含有2个乘号”的例子中，这个范围是2-3。

看到这里，有同学一定会发问，这个公式对于前面的数字也一样符合吗？答案是肯定的。我们可以来测试一下。

举例，现在要求前4个数字中只有1个乘号的最大值。那么根据公式可以得出循环的范围是2-3个数字，乘号的个数为0。查表Dp[2][0]=12x31=372,Dp[3][0]=123x1=123。因此最大值为372，与我们手算的结果相同。

8、对于这个剩余数值，我们可以建立Nums表进行存储。这个表的横向和纵向的概念都是数字的位数，只不过一个代表开头位数，一个代表结束位数。举例Nums[2][4]就代表从第二位开始，到第四位终止的数字，这个值正是231。当然，我们也看得出Nums[2][1]类似这样的数字是无意义的，因此我们用“--”代替。

开头数位i	结束数位j			
	1	2	3	4
1	1	12	123	1231
2	--	2	23	231
3	--	--	3	31
4	--	--	--	1

编码

```
#include <bits/stdc++.h>

using namespace std;

int f_max[11][7], num[11][11];  //动态规划表和剩余数字表
char number[11];
int n, k;

int main() {
    scanf("%d%d", &n, &k);
    getchar();
    //以下的代码是在处理读入的数字
   scanf("%s", number);
    for (int i = 0; i < n; i++) {
        num[i + 1][i + 1] = number[i] - '0';
    }
    for (int j = 2; j <= n; j++) {
        for (int i = j - 1; i >= 1; i--) {
            num[i][j] = num[i][j - 1] * 10 + num[j][j];
        }
    }
    for (int i = 1; i <= n; i++) {
        f_max[i][0] = num[1][i];
    }
    //乘号个数
   for (int l = 1; l <= k; l++) {
        //前几个数
      for (int i = l + 1; i <= n; i++) {
```

```
            //遍历前乘号个数减1时的全部合法情况
            for (int j = l; j <= i - 1; j++) {
                int value = f_max[j][l - 1] * num[j + 1][i];
                f_max[i][l] = max(f_max[i][l], value);
            }
        }
    }
    printf("%d\n", f_max[n][k]);
    return 0;
}
```