

# Problem D - Fight With Courage

## Information

Time Limit	Memory Limit	Source File Name	IO File Name
1000ms	128MiB	delta.cc	delta.in/.out

Problem Type	Data Amount	Points Per Data	Partial Points
Tradition	20	5	Yes

## Description

在那险象环生的重重阻碍之后，我刚好看到了从另一边冲出来的她  
她看起来并无大碍，而我却已是伤痕累累  
如此的“难度差距”，我也只能苦笑以对

.....

除去魔鬼难度的生死闯关外，这里与现实世界并无多少差异  
陌生的街道，陌生的人，极目远眺，就连蓝天白云都是陌生的  
“其实，我也和你一样”

.....

不知道从何时起，一个人的试炼，变成了两个人的历险  
所谓的“引导员”，不过是这个游戏的另一种玩家而已

越是精巧的机关，构件就越多，做工就越精密，但相对的，也就越脆弱；有时只要找出一两个弱点，便能使其崩毁。

身处眼花缭乱的幻境中的你所需要找出的，便是在众多有序排列的可攻击的点中找出包含尽可能多种类弱点的区间，并且这个区间要尽可能的短，以便于击破。

## Input

第一行一个整数 $N$ ，表示总共有 $N$ 个弱点。

第二行到第 $N + 1$ 行，每行一个字符串，表示一个弱点。

第 $N + 2$ 行一个整数 $M$ ，表示总共发现了 $M$ 个可以攻击的点。

第 $N + 3$ 行到第 $N + M + 3$ 行，每行一个字符串，表示一个可以攻击的点。

## Output

第一行一个整数，表示所有可攻击的点中，弱点的种类数。

第二行一个整数，表示所有可攻击的点中，包含弱点种类最多的区间的最小长度。

## Sample Input

```
3
lyy
dzh
xfbp
5
lyy
dzh
dzh
xfbp
lyy
```

## Sample Output

```
3
3
```

## Data Limit

对于30%的数据： $1 \leq N \leq 50, 1 \leq M \leq 500$ 。

对于60%的数据： $1 \leq N \leq 300, 1 \leq M \leq 5000$ 。

对于100%的数据： $1 \leq N \leq 1000, 1 \leq M \leq 100000, 3 \leq \text{length}(\text{string}) \leq 10$ 。

## Attention

所有给定的字符串中仅包含小写字母。

最终答案的区间里，弱点不一定是连续的。

## Tips

对于样例，在所有可攻击的点中，一共出现了3种弱点；最短的包含弱点种类最多的区间是 $[3, 5]$ ，包含了3种弱点，长度为3。

# Problem E - Think With Wisdom

## Information

Time Limit	Memory Limit	Source File Name	IO File Name
1000ms	128MiB	sigma.cc	sigma.in/.out

Problem Type	Data Amount	Points Per Data	Partial Points
Tradition	20	5	No

## Description

没有概念的时光流逝，不知道过了多久  
浑浑噩噩地前进着，感觉全身的关节正渐渐地被锈迹侵蚀

.....  
“我们，为什么一定要前进呢？”  
机械般前进的脚步猛然停下，过往的种种浮现在眼前  
“是啊，一开始就没必要前进的。但是，还有其它的选择么？”

.....  
要想从这个世界级的BUG中逃脱，要么绕过它，要么修复它，要么.....就终止它  
话虽如此，不过，真的做得到吗？  
转身朝向那不断崩毁的道路，我和她都能看到，对方眼中那坚定的光芒

幸运的是，即便看起来已如断壁残垣一般，有些道路还是勉强能通行的，而你手中也有一些助力工具，能帮助你通过一些风险高的道路。

对于整个道路网络，共有 $N$ 个点和 $M$ 条边，你需要从1号点抵达 $N$ 号点，每条道路连接两个点，并有一个风险值 $L_i$ ，表示通过这条道路可能的风险。对于你选择的从1号点通向 $N$ 号点的道路，其风险值等同于路径上所有经过的道路中风险值最大的那条道路的风险值。

你的手中共有 $K$ 个助力工具，每个助力工具只能使用一次，帮助你在无视风险值的情况下通过一条道路。你需要知道的，就是从1号点到 $N$ 号点最小的风险值是多少。

## Input

第一行三个整数 $N, M, K$ 。第二行到第 $M + 1$ 行，每行三个整数 $A_i, B_i, L_i$ ，表示从 $A_i$ 到 $B_i$ 有一条道路，其风险值为 $L_i$ 。

## Output

一行一个整数 $ans$ ，表示从1号点到 $N$ 号点最小的风险值，如果不能到达输出 $-1$ 。

## Sample Input

```
5 7 1
1 2 5
3 1 4
2 4 8
3 2 3
5 2 9
3 4 7
4 5 6
```

# Sample Output

4

## Data Limit

对于100%的数据： $1 \leq N \leq 1000, 1 \leq M \leq 10000, 1 \leq L_i \leq 1000000$ 。

## Attention

如果需要经过的道路 $\leq K$ ，最终风险值就为0，因为一路用助力工具就能过去了。

## Tips

对于样例，选择道路1 - 3 - 2 - 5，风险值= $\max(4, 3, 9)$ ，在最后一道路上使用助力工具无视风险值，最终答案为 $\max(4, 3) = 4$ 。

# Problem F - Escape With Promise

## Information

Time Limit	Memory Limit	Source File Name	IO File Name
1000ms	128MiB	omega.cc	omega.in/.out

Problem Type	Data Amount	Points Per Data	Partial Points
Tradition	20	5	Yes

## Description

似乎这个BUG也没有料到我们会逆流而上，周遭的空间开始变得扭曲起来  
是时候冲破这些虚伪的束缚，回归真实的世界了

.....

“我好像发现了一个漏洞，你保持现状就好。如果片刻之后仍没有动静，你还是回去当你的引导员比较稳妥”

不等她回话，我便转头冲进了扭曲的空间，顿时，意识被撕得粉碎  
在意识消失之前，我感觉到，左臂似乎被什么拉住了

.....

意识回归，视野清晰，还是那个银灰色的房间  
但这次摆在我面前的，并非当年的计数器，而是一台老旧的计算机  
再次回到孤军奋战的状态，不知为何，感觉轻松了些许  
将所有的信念和意志作为赌注，本不该存在的时间竞速开始了

一台老式计算机能干什么？其实它什么都能干，只不过没有先进的编码和图形界面。对于你而言，只要能防守住接下来的数据流攻击，这个虚幻的世界就可以结束了。

为了应对接下来的数据流攻击，你需要对你手中不多的防御资源进行良好的调度，以确保防御体系的完整。

有两种防御资源可供调度：自动迎击模块A和自我修复模块B，两种模块总共有 $N$ 种，并按顺序排列在硬盘的扇区内等待调用。

你需要建立一些防御阵列，每个防御阵列可以包含任意多的两类模块，但是由于数据流攻击的不确定性，每个防御阵列中要么只有一种模块，要么两种模块的数量差不能超过 $K$ ，否则就有被攻破的风险。然而，由于操作系统过于老旧，你能执行的操作有诸多限制，所以你能只能将连续一段硬盘扇区内的模块划分到一个防御阵列中。

你要做的就是求出最少的需要划分出的防御阵列的数量，以尽可能地减少操作量，增强防御效果。

## Input

第一行两个整数 $N, K$ ，表示有 $N$ 个扇区，单一防御阵列中两种模块的数量差不能超过 $K$ ；每个扇区内仅包含一个模块。

第二行到第 $N + 1$ 行，每行一个字符 $C_i$ ，表示第 $i$ 个扇区内包含的模块的种类。

## Output

一行一个整数 $ans$ ，表示最少的需要划分出的防御阵列的数量。

## Sample Input

```
5 1
A
A
B
B
A
```

## Sample Output

```
1
```

## Data Limit

对于15%的数据： $N \leq 2500, K \leq 10$ 。

对于30%的数据： $N \leq 500000, K \leq 10$ 。

对于100%的数据： $N \leq 500000, K \leq 2000$ 。

## Attention

注意每个防御阵列包含的所有模块在给定的扇区序列中必须是 **连续分布** 的。

## Tips

对于样例，由于两种模块的数量差为1，所以全部划分到一个防御阵列里即可。