

Claris' Contest # 2

Day 1

By Claris

September 3, 2016

题目名称	String Master	Tourist Attractions	Walk
输入文件名	master.in	tour.in	walk.in
输出文件名	master.out	tour.out	walk.out
每个测试点时限	1s	1s	1s
测试点数目	10	10	10
每个测试点分值	10	10	10
内存限制	128MB	128MB	128MB
是否有部分分	否	否	否
题目类型	传统	传统	传统

Problem A. String Master(master.c/cpp/pas)

Input file: `master.in`
Output file: `master.out`
Time limit: 1 seconds
Memory limit: 128 megabytes

所谓最长公共子串，比如串 A: “abcde”，串 B: “jcdkl”，则它们的最长公共子串为串 “cd”，即长度最长的字符串，且在两个串中都作为连续子串出现过。

给定两个长度都为 n 的字符串，对于字符串大师的你来说，求它们的最长公共子串再简单不过了。所以现在你有 k 次修改机会，每次你可以选择其中某个串的某个位置，将其修改成任意字符。

你需要合理使用这 k 次修改机会，使得修改之后两个串的最长公共子串最长。相信对于字符串大师的你来说，这个问题也难不倒你。

Input

第一行包含两个整数 n, k ，分别表示字符串的长度和修改次数。

第二行包含一个长度为 n 的仅由小写字母构成的字符串 S 。

第三行包含一个长度为 n 的仅由小写字母构成的字符串 T 。

Output

输出一行一个整数，即修改完毕之后两个串的最长公共子串的长度。

Examples

<code>master.in</code>	<code>master.out</code>
5 0 abcde jcdkl	2
5 2 aaaaa ababa	5

Notes

对于 100% 的数据, $0 \leq k \leq n$ 。

测试点编号	n	k
1	= 5	= 0
2	= 10	= 0
3	= 10	= 0
4	= 10	= 1
5	= 10	= 1
6	= 100	≤ 100
7	= 150	≤ 150
8	= 200	≤ 200
9	= 250	≤ 250
10	= 300	≤ 300

Problem B. Tourist Attractions(tour.c/cpp/pas)

Input file: tour.in
 Output file: tour.out
 Time limit: 1 seconds
 Memory limit: 128 megabytes

在美丽的比特镇一共有 n 个景区，编号依次为 1 到 n ，它们之间通过若干条双向道路连接。

Byteasar 慕名来到了比特镇旅游，不过由于昂贵的门票费，他只能负担起 4 个景区的门票费。他可以在任意景区开始游览，然后结束在任意景区。

Byteasar 的旅游习惯比较特殊，一旦他路过了一个景区，他就一定会进去参观，并且他永远不会参观同一个景区两次。所以他想知道，有多少种可行的旅游路线，使得他可以恰好参观 4 个景区呢？即，有多少条简单路径恰好经过了 4 个点。

Input

第一行包含两个整数 n ，表示景区的总数。

第 2 至第 $n+1$ 行，每行一个长度为 n 的 01 字符串，第 $i+1$ 行第 j 个字符为 0 表示 i 和 j 之间没有道路，为 1 表示有一条道路。

输入数据保证 (i, j) 的连接情况等于 (j, i) 的连接情况，且 (i, i) 恒为 0。

Output

输出一行一个整数，即可行的路线总数。

Examples

tour.in	tour.out
4 0101 1010 0101 1010	8

8 条路线分别为：

1->2->3->4, 4->3->2->1,

2->3->4->1, 1->4->3->2,

3->4->1->2, 2->1->4->3,

4->1->2->3, 3->2->1->4。

Notes

测试点编号	n
1	= 5
2	= 10
3	= 20
4	= 50
5	= 300
6	= 300
7	= 300
8	= 1500
9	= 1500
10	= 1500

Problem C. Walk(walk.c/cpp/pas)

Input file: walk.in
 Output file: walk.out
 Time limit: 1 seconds
 Memory limit: 128 megabytes

在比特镇一共有 n 个街区，编号依次为 1 到 n ，它们之间通过若干条单向道路连接。

比特镇的交通系统极具特色，除了 m 条单向道路之外，每个街区还有一个编码 val_i ，不同街区可能拥有相同的编码。如果 val_i and $val_j = val_j$ ，即 val_i 在二进制下与 val_j 做与运算等于 val_j ，那么也会存在一条额外的从 i 出发到 j 的单向道路。

Byteasar 现在位于 1 号街区，他想知道通过这些道路到达每一个街区最少需要多少时间。因为比特镇的交通十分发达，你可以认为通过每条道路都只需要 1 单位时间。

Input

第一行包含两个正整数 n, m ，表示街区的总数以及道路的总数。

第二行包含 n 个正整数 $val_1, val_2, \dots, val_n$ ，分别表示每个街区的编码。

接下来 m 行，每行包含两个正整数 u_i, v_i ，表示一条单向道路，起点为 u_i ，终点为 v_i 。

Output

输出 n 行，每行一个整数，其中第 i 行输出到达第 i 个街区的最少时间，如果无法到达则输出 -1 。

Examples

walk.in	walk.out
5 2	0
5 4 2 3 7	1
1 4	2
2 3	1
	-1

Notes

对于 100% 的数据, $1 \leq u_i, v_i \leq n, 1 \leq val_i < 2^{20}$ 。

测试点编号	n	m	val_i
1	= 5	≤ 10	$< 2^4$
2	= 5	≤ 10	$< 2^4$
3	= 2000	≤ 5000	$< 2^{10}$
4	= 2000	≤ 5000	$< 2^{10}$
5	= 200000	≤ 300000	$< 2^{15}$
6	= 200000	≤ 300000	$< 2^{15}$
7	= 200000	≤ 300000	$< 2^{15}$
8	= 200000	≤ 300000	$< 2^{20}$
9	= 200000	≤ 300000	$< 2^{20}$
10	= 200000	≤ 300000	$< 2^{20}$