

问题 A: 环形城市

zz 住在一个环形城市里，城市里有 n 栋房子。环路房屋按顺时针顺序编号为 1 到 n 。环路交通是单向的，只能顺时针走。

zz 最近搬进了环路 1 号房子，去完成 m 个任务。zz 只能按顺序完成各项任务，为了完成第 i 个任务，她需要在 a_i 号房子里完该任务。最初，zz 在 1 号房子里，他只能顺时针沿着环路走。从一个房子走到相邻的房子需要一个单位的时间，那么他完成所有任务所需的最短时间。

输入格式

第一行包含两个整数 n 和 m ($2 \leq n \leq 10^5, 1 \leq m \leq 10^5$)。第二行包含 m 个整数 a_1, a_2, \dots, a_m ($1 \leq a_i \leq n$)。请注意，zz 可以在一个房子里完成多个连续的任务。

输出格式

输出一个整数——zz 完成所有任务所需的时间。

建议使用 `long long` 定义，最好使用 `cin`、`cout`。

输入	输出
4 3 3 2 3	6
4 3 2 3 3	2

Note

在第一个测试示例中，zz 沿着环路移动的顺序如下。: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 3$ 。这是最佳顺序。所以，她需要 6 个时间单位。

问题 B: 自然数

自然数 1 到 N ，按顺序列成一排，你可以从中取走任意个数，但是相邻的两个不可以同时被取走。计算一共有多少种取法。

输入格式

仅包含一个数 n ($1 < n < 50$)。

输出格式

仅包含一个数——你的答案。

输入	输出
5	13

数据范围与提示

$n=45$ 的时候，已经超过 1 亿，用 `long long int` 定义

问题 C: 泰迪

ZZ 拥有 N 头有斑点的泰迪和 N 头没有斑点的泰迪。他刚刚完成了泰迪遗传学课程，他确信泰迪上的斑点是由泰迪基因组突变引起的。

ZZ 花了大价钱对他的泰迪的基因组进行了测序。每个基因组都是一个由四个字符 A,C,G,T 构成的长度为 M 的字符串。当他统计得到的泰迪的基因组序列时，可以得到一个如下所示的表：（此时， $N=3$ ）



位置: 1 2 3 4 5 6 7 ... M

斑点泰迪 1: AATCCCA... T

斑点泰迪 2: GATTGCA... A

斑点泰迪 3: GGTCGCA... A

普通泰迪 1: ACTCCCA... G

普通泰迪 2: AGTTGCA... T

普通泰迪 3: AGTTCCA... T

通过仔细观察该表，他发现通过位置 2 和位置 4 的字符足以判断泰迪是否存在斑点。也就是说，仅通过查看这两个位置上的字符，ZZ 就可以判断他的哪些泰迪有斑点，哪些没有斑点。

例如，如果他看到 G 和 C，他就知道那头泰迪一定是有斑点的。

ZZ 坚信是否存在斑点不仅仅可以通过基因组中的一个或两个位置来判断，还可以通过观察三个不同的位置来做出判断。

请帮助他求出共有多少组三个不同的位置可以用来判断是否存在斑点。

输入格式

第一行包含整数 N 和 M 。接下来 N 行，每行包含一个长度为 M 的字符串，用来描述斑点泰迪的基因组序列。再接下来 N 行，每行包含一个长度为 M 的字符串，用来描述普通泰迪的基因组序列。

输出格式

输出可以用来判断是否存在斑点的三个不同位置的组数。

数据范围

$1 \leq N \leq 500, 3 \leq M \leq 50$

输入	输出
3 8 AATCCCAT GATTGCAA GGTCGCAA ACTCCAG ACTCGCAT ACTTCCAT	22

问题 D: 迷宫

有一个 $M \times N$ 的迷宫，每个点 (x,y) 都是一个房间。zz 从入口 $(0,0)$ 到了房间 $(M-1,N-1)$ 。zz 逃出迷宫的唯一办法就是只经过他来时所经过的那些房间，到达 $(0,0)$ ，且他每次在不出迷宫的前提下，可以从 (x,y) 移动到 $(x-1,y),(x-2,y),(x,y-2),(x,y-1)$ 四个房间中的一个。

但是他已经忘记了他来时经过了哪些点，他只知道来时他在不出迷宫的前提下每次可以从 (x,y) 移动到 $(x+1,y),(x+2,y),(x,y+1),(x,y+2)$ 。

求 zz 从 $(0,0)$ 到 $(M-1,N-1)$ 再回到 $(0,0)$ ，可能的路径总数，对 10^9+7 取模。题目有 T 组数据。

$1 \leq T \leq 101 \leq T \leq 10$ ， $2 \leq M, N \leq 500$ 。

输入格式

第一行包含 T ，即测试用例的数量。

每个测试用例包含 2 个空格分隔的整数 M 和 N 。

输出格式

输出 T 行，每行包含所需答案模 10^9+7 。

输入	输出
3	2
2 2	7
2 3	66
3 4	

对于第二组数据，一种可能的路径如下：

$(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (0,2) \rightarrow (0,0)$ 。

问题 E: 寻找泰迪

一向以技术娴熟著称的 ZZ 正在测试他的新型自动无人驾驶泰迪定位摄像机，并自动找出泰迪的位置。不幸的是，相机没有很好的寻找泰迪的算法，因此 ZZ 需要你的帮助来开发出更好的算法。

泰迪可用一个 $N \times N$ 的字符矩阵来描述，每个字符在 A...Z 范围内，代表 26 种可能的颜色之一。

现给出“潜在泰迪位置 (PCL)”的定义：

PCL 是一个图像矩阵的子矩阵（也可能是整个图像矩阵），其边与图像边平行，且不包含在任何其他 PCL 中（因此，PCL 的较小子集也不是 PCL）。

此外，PCL 必须满足以下属性：PCL 矩阵内部恰好包含两种颜色，一种颜色构成一个连通区域，另一种颜色构成两个或更多个连通区域。

例如，一个矩阵图像如下：

AAAAA

ABABA

AAABB

可以视为一个 PCL 矩阵，因为 A 构成单个连通区域，B 构成多个连通区域。这可以解释为是一头 A 色的泰迪，身上长着 B 色的斑点。

如果从一个区域内的任意一个位置出发，仅通过向上，向下，向左，向右的方式逐步移动，可以到达区域内任意其他位置，则称这个区域是连通的。

给定图像矩阵，请你找出其中 PCL 的数量。

输入第一行包含整数 $N(1 \leq N \leq 20)$ 。接下来 N 行，描述图像矩阵，每行包含 N 个字符。

输出给定图像中，PCL 的数量。

4 ABBC BBBC AABB ABBC	2
4 ABBA CBAD BBBB CADA	4

在第一个样例中，共有 2 个 PCL 矩阵，分别是：

ABB	BC
BBB	BC
AAB	BB
ABB	BC

程序阅读题（90分）：在 OJ 上提交答案，我人工判题。选择题 10 分一题，判断题 5 分一题。

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4
5 using namespace std;
6
7 int f(const string &s, const string &t)
8 {
9     int n = s.length(), m = t.length();
10
11     vector<int> shift(128, m + 1);
12
13     int i, j;
14
15     for (j = 0; j < m; j++)
16         shift[t[j]] = m - j;
17
18     for (i = 0; i <= n - m; i += shift[s[i + m]]){
19         j = 0;
20         while(j < m && s[i + j] == t[j]) j++;
21         if (j == m) return i;
22     }
23
24     return -1;
25 }
26
27 int main()
28 {
29     string a ,b;
30     cin >> a >> b;
31     cout << f(a, b) << endl;
32     return 0;
33 }
```

假设输入字符串由 ASCII 可见字符组成，完成下面的判断题和单选题：

判断题

1. (1 分) 当输入为 “abcde fg” 时，输出为-1。
2. 当输入为 “abbababbbab abab” 时，输出为 4。
3. 当输入为 “GoodLuckCsp2022 22” 时，第 20 行的 “j++” 语句执行次数为 2。

单选题

4. 该算法最坏情况下的时间复杂度为 ()。
5. **f(a, b)** 与下列 () 语句的功能最类似。
6. 当输入为 “baaabaaabaaabaaaa aaaa” ，第 20 行的 “j++” 语句执行次数为 ()。

1. A. 正确
 B. 错误
2. A. 正确
 B. 错误
3. A. 正确
 B. 错误
4. A. $O(n + m)$
 B. $O(n \log m)$
 C. $O(m \log n)$
 D. $O(nm)$
5. A. a.find(b)
 B. a.rfind(b)
 C. a.substr(b)
 D. a.compare(b)
6. A. 9
 B. 10
 C. 11
 D. 12

```

1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 105;
6
7  int n, m, k, val[MAXN];
8  int temp[MAXN], cnt[MAXN];
9
10 void init()
11 {
12     cin >> n >> k;
13     for (int i = 0; i < n; i++) cin >> val[i];
14     int maximum = val[0];
15     for (int i = 1; i < n; i++)
16         if (val[i] > maximum) maximum = val[i];
17     m = 1;
18     while (maximum >= k) {
19         maximum /= k;
20         m++;
21     }
22 }
23
24 void solve()
25 {
26     int base = 1;
27     for (int i = 0; i < m; i++) {
28         for (int j = 0; j < k; j++) cnt[j] = 0;
29         for (int j = 0; j < n; j++) cnt[val[j] / base % k]++;
30         for (int j = 1; j < k; j++) cnt[j] += cnt[j - 1];
31         for (int j = n - 1; j >= 0; j--) {
32             temp[cnt[val[j] / base % k] - 1] = val[j];
33             cnt[val[j] / base % k]--;
34         }
35         for (int j = 0; j < n; j++) val[j] = temp[j];
36         base *= k;
37     }
38 }
39
40 int main()
41 {
42     init();
43     solve();
44     for (int i = 0; i < n; i++) cout << val[i] << ;
45     cout << endl;
46     return 0;
47 }

```

假设输入的 n 为不大于 100 的正整数, k 为不小于 2 且不大于 100 的正整数, $val[i]$ 在 int 表示范围内, 完成下面的判断题和单选题:

判断题

1. 这是一个不稳定的排序算法。 ()
2. 该算法的空间复杂度仅与 n 有关。 ()
3. 该算法的时间复杂度为 $O(m(n + k))$ 。 ()

单选题

4. 当输入为 "5 3 98 26 91 37 46" 时, 程序第一次执行到第 36 行, $val[]$ 数组的内容依次为 ()。
 5. 若 $val[i]$ 的最大值为 100, k 取 () 时算法运算次数最少。
 6. 当输入的 k 比 $val[i]$ 的最大值还大时, 该算法退化为 () 算法。
1. A. 正确
 B. 错误
 2. A. 正确
 B. 错误
 3. A. 正确
 B. 错误
 4. A. 91 26 46 37 98
 B. 91 46 37 26 98
 C. 98 26 46 91 37
 D. 91 37 46 98 26
 5. A. 2
 B. 3
 C. 10
 D. 不确定
 6. A. 选择排序
 B. 冒泡排序
 C. 计数排序
 D. 桶排序