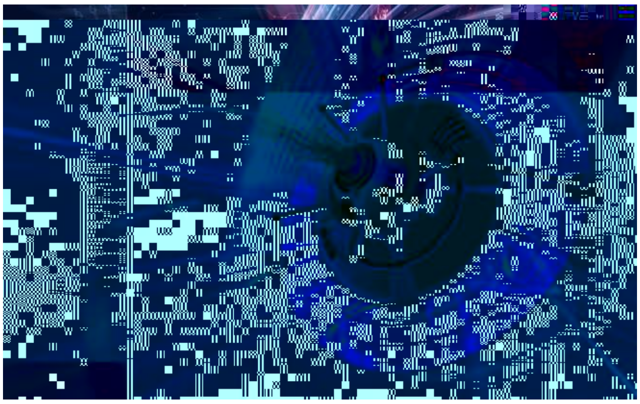
题目名称	暴走的图灵机	奇妙的数列	染色配对
程序文件名	bugs	array	pair
输入文件名	bugs.in	array.in	pair.in
输出文件名	bugs.out	array.out	pair.out
每个测试点时限	1 秒	1 秒	1 秒
内存限制	256MB	256MB	256MB
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	无
试题类型	传统	传统	传统
提交源程序须加后缀			
对于 Pascal 语言	bugs.pas	array.pas	pair.pas
对于 C 语言	bugs.c	array.c	pair.c
对于 C++语言	bugs.cpp	array.cpp	pair.cpp

暴走的图灵机

(bugs.pas/c/cpp)

题目描述



所谓的图灵机就是指一个抽象的机器，它有一条无限长的纸带，纸带分成了一个一个的小方格，每个方格有不同的颜色。有一个机器头在纸带上移来移去。机器头有一组内部状态，还有一些固定的程序。在每个时刻，机器头都要从当前纸带上读入一个方格信息，然后结合自己的内部状态查找程序表，根据程序输出信息到纸带方格上，并转换自己的内部状态，然后进行移动。

一种最朴素的图灵机只有 0 和 1，一种高度抽象的图灵机初始的内部状态是这样表示的：“我的左手是字符串”0”，右手是字符串”1”。”之后图灵机会进行 N 次操作，每次操作之后，图灵机的左手将会变成操作前右手的字符串，而图灵机的右手将会变成操作前左右手从左至右链接起来的字符串。

人类对一个高度抽象的图灵机是否满意取决于图灵机的 N 次操作后，左手的字符串包含多少个已知子串 T。为了简化问题，你只需计算这种图灵机 N 次操作后所包含的 T 串个数除以 P 的余数。

输入格式

第一行有三个正整数 N、M、P，N、P 如题中所述，M 表示串 T 的长度。

第二行为一个长度为 M 的 01 串 T。

输出格式

仅包含一个整数，表示答案除以 P 的余数

样例输入

7 3 100

101

样例输出

8

数据范围和注释

对于 30%的数据，N≤20。

对于 60%的数据，N≤10^5，M≤200。

对于 100%的数据，N≤10^9，M≤10000，P≤10^9

奇妙的数列

(array.pas/c/cpp)

题目描述

传说古希腊毕达哥拉斯（约公元前 570-约公元前 500 年）学派的数学家经常在沙滩上研究数学问题，他们在沙滩上画点或用小石子来表示数。比如，他们研究过三角形点阵表示，1,3,6,10,15,21...，他们就将其称为三角形数。
类似地，1,4,9,16,25,36...被称为正方形数，因为这些数能够表示成正方形。
因此，按照一定顺序排列的一列数称为数列。
相传，瑞士数学家、自然科学家莱昂哈德•欧拉也研究过一种奇妙的有穷数列 A，它是由另一个有穷数列 B 导出的。数列 A 性质奇特，虽然每一项与相邻几项并无直接关系，但是若将数列 A 整体看来，却与 B 有着宏观上的近似之处。它具体表现为 $a_n=n-k+1$ 其中 k 为最小的正整数使得数列 B 中，对于所有 $k<=i<=n$ 的 i 均满足 $b_k<=b_i<=b_n$。
我们并不关心欧拉是如何逐项计算 A 的，但我们颇感兴趣的问题是 A 中的最大值是多少。

输入格式

第一行为一个正整数 N 表示已知的数列 B 共有 N 项。
第二行有 N 个数依次表示 $b_1,b_2,b_3,b_4,...,b_n$。

输出格式

仅包含一个整数，表示数列 A 中的最大值。

样例输入

9
2 1 6 5 4 2 7 2 2

样例输出

6

数据范围和注释

对于 20%的数据，N≤5000。
对于 50%的数据，N≤10^5。
对于 100%的数据，N≤10^7。
对于 30%为随机数列。

提示：由于输入数据过大，请注意读入优化

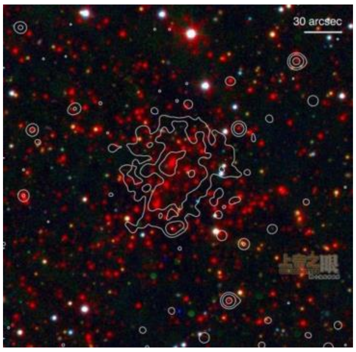
```
inline int ReadInt(){
    char ch = getchar();
    int data = 0;
    while (ch < '0' || ch > '9')
        ch = getchar();
    do
    {
        data = data*10 + ch-'0';
        ch = getchar();
    } while (ch >= '0' && ch <= '9');
    return data;
}
```

染色配对

(pair.pas/c/cpp)

题目描述

图匹配是图论(Graph Theory)研究的重要方向之一。当前研究较为完善的是对二分图这一特殊模型计算其极大匹配(Maximal Matching)，所用的通常是匈牙利算法(Hungarian Algorithm)或网络流(Maximal Flow)。而对于一般图的极大匹配目前研究较为完善的有带花树算法又称埃德蒙通用匹配算法(Edmonds’s matching algorithm)，其时间复杂度与匈牙利算法同样是 O(VE)。



图匹配研究过程中，有学者指出对于具有特殊性质的一般图匹配问题会有更加低的时间复杂度求出极大匹配。如染色匹配问题(Dyed matching problem)，其表现为图中的每个结点均恰好属于两个极大团(每个极大团之内的结点两两均有边联结，且极大团不被任何一个除自己本身之外的团所包含)，以上性质称为图的可染色性。

我们的目标就是计算一个具有染色性质的图的极大匹配并提供匹配方案。

输入格式

第一行为两个正整数 M,N，表示共有 M 个极大团，图中共有 N 个结点。
接下来 N 行，每行有 2 个整数 a,b，依次表示每个结点所属的两个极大团的编号为 a,b(保证 a≠b 且 a,b≤M)，结点编号从 1 开始。

输出格式

第一行输出极大匹配个数 K，接下来输出一个合法的匹配方案，共 K 行，每行两个正整数 a,b 表示每组匹配结点的编号。

样例输入

```
7 10
1 2
1 3
2 4
3 4
4 5
4 6
5 7
6 7
2 5
3 6
```

样例输出

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

4 10

数据范围和注释

对于 20%的数据，N,M≤10。
对于 40%的数据，M≤100，N≤1000。
对于 60%的数据，M≤1000，N≤10000。
对于 100%的数据，M≤20000，N≤200000。