

# CCF NOI 2018 省队选拔赛

## 第一试

时间：2018 年 4 月 6 日 08:00 ~ 12:30

题目名称	一双木棋	IIIDX	秘密袭击
题目类型	传统型	传统型	传统型
目录	chess	iiidx	coat
可执行文件名	chess	iiidx	coat
输入文件名	chess.in	iiidx.in	coat.in
输出文件名	chess.out	iiidx.out	coat.out
每个测试点时限	1.0 秒	2.0 秒	5.0 秒
内存限制	512 MB	512 MB	1 GB
测试点数目	20	20	20
每个测试点分值	5	5	5

提交源程序文件名

对于 C++ 语言	chess.cpp	iiidx.cpp	coat.cpp
对于 C 语言	chess.c	iiidx.c	coat.c
对于 Pascal 语言	chess.pas	iiidx.pas	coat.pas

编译选项

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11
对于 Pascal 语言	-O2	-O2	-O2

## 一双木棋 (chess)

### 【题目描述】

菲菲和牛牛在一块  $n$  行  $m$  列的棋盘上下棋，菲菲执黑棋先手，牛牛执白棋后手。

棋局开始时，棋盘上没有任何棋子，两人轮流在格子上落子，直到填满棋盘时结束。落子的规则是：一个格子可以落子当且仅当这个格子内没有棋子且这个格子的左侧及上方的所有格子内都有棋子。

棋盘的每个格子上，都写有两个非负整数，从上到下第  $i$  行中从左到右第  $j$  列的格子上的两个整数记作  $A_{ij}$ 、 $B_{ij}$ 。在游戏结束后，菲菲和牛牛会分别计算自己的得分：菲菲的得分是所有有黑棋的格子上的  $A_{ij}$  之和，牛牛的得分是所有有白棋的格子上的  $B_{ij}$  的和。

菲菲和牛牛都希望，自己的得分减去对方的得分得到的结果最大。现在他们想知道，在给定的棋盘上，如果双方都采用最优策略且知道对方会采用最优策略，那么，最终的结果如何。

### 【输入格式】

从文件 *chess.in* 中读入数据。

输入第一行包含两个正整数  $n, m$ ，保证  $n, m \leq 10$ 。

接下来  $n$  行，每行  $m$  个非负整数，按从上到下从左到右的顺序描述每个格子上的第一个非负整数：其中第  $i$  行中第  $j$  个数表示  $A_{ij}$ 。

接下来  $n$  行，每行  $m$  个非负整数，按从上到下从左到右的顺序描述每个格子上的第二个非负整数：其中第  $i$  行中第  $j$  个数表示  $B_{ij}$ 。

### 【输出格式】

输出到文件 *chess.out* 中。

输出一个整数，表示菲菲的得分减去牛牛的得分的结果。

### 【样例 1 输入】

```
2 3
2 7 3
9 1 2
3 7 2
2 3 1
```

## 【样例 1 输出】

2

## 【样例 1 解释】

2	7	3
3	7	2
9	1	2
2	3	1

棋盘如图所示，双方都采用最优策略时，棋局如下：

- 菲菲下在第 1 行第 1 列（这是第一步时唯一可以落子的格子）；
- 牛牛下在第 1 行第 2 列；
- 菲菲下在第 2 行第 1 列；
- 牛牛下在第 1 行第 3 列；
- 菲菲下在第 2 行第 2 列；
- 牛牛下在第 2 行第 3 列（这是这一步时唯一可以落子的格子）；
- 填满棋盘，游戏结束，盘面如下。

2 ●	7 ○	3 ○
3	7	2
9 ●	1 ●	2 ○
2	3	1

菲菲的得分为： $2 + 9 + 1 = 12$ ；牛牛的得分为： $7 + 2 + 1 = 10$ 。

## 【样例 2】

见选手目录下的 *chess/chess2.in* 与 *chess/chess2.ans*。

**【子任务】**

对于所有的测试数据， $n, m \leq 10$ ， $A_{ij}, B_{ij} \leq 100000$ 。

对于编号为奇数的测试点，保证所有的  $B_{ij} = 0$ 。

测试点	$n =$	$m =$
1, 2, 3	2	2
4, 5, 6	3	3
7, 8	5	5
9, 10	8	8
11, 12	10	1
13, 14	10	2
15, 16	10	3
17, 18, 19, 20	10	10

## IIDX (iidx)

## 【题目背景】

Osu 听过没？那是 Konano 最喜欢的一款音乐游戏，而他的梦想就是有一天自己也能做个独特酷炫的音乐游戏。现在，他在世界知名游戏公司 KONMAI 内工作，离他的梦想也越来越近了。

这款音乐游戏内一般都包含了许多歌曲，歌曲越多，玩家越不易玩腻。同时，为了使玩家在游戏上氪更多的金钱花更多的时间，游戏一开始一般都不会将所有曲目公开，有些曲目你需要通关某首特定歌曲才会解锁，而且越晚解锁的曲目难度越高。

## 【题目描述】

这一天，Konano 收到了一个任务，他需要给正在制作中的游戏《IIDX》安排曲目的解锁顺序。游戏内共有  $n$  首曲目，每首曲目都会有一个难度  $d$ ，游戏内第  $i$  首曲目会在玩家 Pass 第  $\lfloor \frac{i}{k} \rfloor$  首曲目后解锁（ $\lfloor x \rfloor$  为下取整符号）若  $\lfloor \frac{i}{k} \rfloor = 0$ ，则说明这首曲目无需解锁。

举个例子：当  $k = 2$  时，第 1 首曲目是无需解锁的（ $\lfloor \frac{1}{2} \rfloor = 0$ ），第 7 首曲目需要玩家 Pass 第  $\lfloor \frac{7}{2} \rfloor = 3$  首曲目才会被解锁。

Konano 的工作，便是安排这些曲目的顺序，使得每次解锁出的曲子的难度不<sub>低</sub>于作为条件需要玩家通关的曲子的难度，即使得确定顺序后的曲目的难度对于每个  $i$  满足  $d_i \geq d_{\lfloor \frac{i}{k} \rfloor}$ 。

当然这难不倒曾经在信息学竞赛摸鱼许久的 Konano。那假如是你，你会怎么解决这份任务呢？

## 【输入格式】

从文件 *iidx.in* 中读入数据。

第 1 行 1 个正整数  $n$  和 1 个小数  $k$ ， $n$  表示曲目数量， $k$  其含义如题所示。

第 2 行  $n$  个用空格隔开的正整数  $d$ ，表示这  $n$  首曲目的难度。

## 【输出格式】

输出到文件 *iidx.out* 中。

输出 1 行  $n$  个整数，按顺序输出安排完曲目顺序后第  $i$  首曲目的难度。

若有多解，则输出  $d_1$  最大的；若仍有多解，则输出  $d_2$  最大的，以此类推。

## 【样例 1 输入】

```
4 2.0
114 514 1919 810
```

## 【样例 1 输出】

```
114 810 514 1919
```

## 【子任务】

测试点编号	$n$	$k$	$d$	特殊限制		
1	$1 \leq n \leq 10$	$k = 2$	$1 \leq d \leq 100$	保证 $d_i$ 互不相同		
2		$k = 3$				
3		$k = 1.1$				
4		$k = n$				
5		$1 < k \leq 100$				
6						
7	$1 \leq n \leq 2000$	$k = 2$	$1 \leq d \leq 10^9$	保证 $d_i$ 互不相同		
8		无				
9		$k = 3$		保证 $d_i$ 互不相同		
10		无				
11		$1 < k \leq 10^9$		保证 $d_i$ 互不相同		
12				无		
13	$1 \leq n \leq 500000$	$k = 2$	$1 \leq d \leq 10^9$	无		
14		$k = 3$		保证 $d_i$ 互不相同		
15		$1 < k \leq 10^9$		保证 $d_i$ 互不相同		
16				无		
17				无		
18				无		
19						
20						

## 秘密袭击 (coat)

### 【题目背景】

*We could have had it all.....*

我们本该，拥有一切

*Counting on a tree.....*

何至于此，数数树上

Counting on a Tree (CoaT) 即是本题的英文名称。

### 【题目描述】

Access Globe 最近正在玩一款战略游戏。在游戏中，他操控的角色是一名 C 国士兵。他的任务就是服从指挥官的指令参加战斗，并在战斗中取胜。

C 国即将向 D 国发动一场秘密袭击。作战计划是这样的：选择 D 国的  $s$  个城市，派出 C 国战绩最高的  $s$  个士兵分别秘密潜入这些城市。每个城市都有一个危险程度  $d_i$ ，C 国指挥官会派遣战绩最高的士兵潜入所选择的城市中危险程度最高的城市，派遣战绩第二高的士兵潜入所选择的城市中危险程度次高的城市，以此类推（即派遣战绩第  $i$  高的士兵潜入所选择城市中危险程度第  $i$  高的城市）。D 国有  $n$  个城市， $n-1$  条双向道路连接着这些城市，使得这些城市两两之间都可以互相到达。为了任务执行顺利，C 国选出的  $s$  个城市中，任意两个所选的城市，都可以不经过未被选择的城市互相到达。

Access Globe 操控的士兵的战绩是第  $k$  高，他希望能估计出最终自己潜入的城市危险程度。Access Globe 假设 C 国是以等概率选出任意满足条件的城市集合  $S$ ，他希望你帮他求出所有可能的城市集合中，Access Globe 操控的士兵潜入城市的危险程度之和。如果选择的城市不足  $k$  个，那么 Access Globe 不会被派出，这种情况下危险程度为 0。

当然，你并不想帮他解决这个问题，你也不打算告诉他这个值除以 998,244,353 的余数，你只打算告诉他这个值除以 64,123 的余数。

### 【输入格式】

从文件 *coat.in* 中读入数据。

第 1 行包含 3 个整数  $n$ 、 $k$ 、 $W$ ，表示 D 国城市的个数、Access Globe 所操控士兵潜入的城市战绩排名以及 D 国的所有城市中最大的危险程度；

第 2 行包含  $n$  个 1 到  $W$  之间的整数  $d_1, d_2, \dots, d_n$ ，表示每个城市的危险程度；

第 3 行到第  $n+1$  行，每行两个整数  $x_i, y_i$ ，表示 D 国存在一条连接城市  $x_i$  和城市  $y_i$  的双向道路。

**【输出格式】**

输出到文件 *coat.out* 中。

输出一个整数，表示所有可行的城市集合中，Access Globe 操控的士兵潜入城市的危险程度之和除以 64,123 的余数。

**【样例 1 输入】**

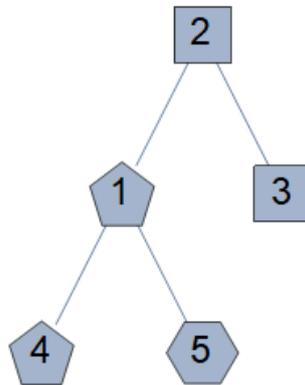
```
5 3 3
2 1 1 2 3
1 2
2 3
1 4
1 5
```

**【样例 1 输出】**

```
11
```

**【样例 1 解释】**

D 国地图如下，其中危险程度为  $d$  的城市的形状是  $(d + 3)$  边形。



以下是所有符合条件且选择的城市不少于 3 个的方案：

- 选择城市 1、2、3，Access Globe 的士兵潜入的城市危险程度为 1；
- 选择城市 1、2、3、4，Access Globe 的士兵潜入的城市危险程度为 1；
- 选择城市 1、2、3、5，Access Globe 的士兵潜入的城市危险程度为 1；
- 选择城市 1、2、3、4、5，Access Globe 的士兵潜入的城市危险程度为 2；
- 选择城市 1、2、4，Access Globe 的士兵潜入的城市危险程度为 1；
- 选择城市 1、2、5，Access Globe 的士兵潜入的城市危险程度为 1；

- 选择城市 1、2、4、5，Access Globe 的士兵潜入的城市危险程度为 2；
- 选择城市 1、4、5，Access Globe 的士兵潜入的城市危险程度为 2；而在选择的

城市少于 3 时，Access Globe 的士兵潜入的城市危险程度均为 0；  
所以你应该输出  $(1 + 1 + 1 + 2 + 1 + 1 + 2 + 2) \bmod 64,123 = 11$ 。

### 【样例 2 输入】

```
10 2 3
2 1 1 3 1 2 3 3 1 3
1 2
2 3
2 4
2 5
2 6
5 7
1 8
8 9
1 10
```

### 【样例 2 输出】

```
435
```

## 【子任务】

测试点	$n =$	$k \leq$	$W =$	地图是一条链
1	3	$n$	1,666	保证
2	15			不保证
3	20		10	保证
4	300			1,666
5	1,666			
6	50			
7	200			
8	500		$10^2$	$10^2$
9	1,111	50		
10			$10^2$	1,666
11	$n$			
12				
13				
14				
15	1,666	$n$	1,666	
16				
17				
18				
19				
20				

对于所有的数据,  $1 \leq k \leq n$ ,  $1 \leq d_i \leq W$ ,  $n, k, W \leq 1,666$ 。