

# 黄冈中学信息奥赛训练题

测试时间 8:30—12:00

(请仔细阅读本页面内容)

## 一. 题目概况

| 中文题目名称    | 搭积木               | 城堡         | 跳跃       |
|-----------|-------------------|------------|----------|
| 英文题目与子目录名 | box               | castle     | jump     |
| 可执行文件名    | box               | castle     | jump     |
| 输入文件名     | box.in            | castle.in  | jump.in  |
| 输出文件名     | box.out           | castle.out | jump.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       |
| 附加样例文件    | 有                 | 有          | 有        |
| 结果比较方式    | 全文比较（过滤行末空格及文末换行） |            |          |
| 题目类型      | 传统                | 传统         | 传统       |
| 运行内存上限    | 256M              | 256M       | 256M     |

## 二. 提交源程序文件名





|           |         |            |          |
|-----------|---------|------------|----------|
| 对于 C++ 文件 | box.cpp | castle.cpp | jump.cpp |
| 对于 C 文件   | box.c   | castle.c   | jump.c   |

## 三. 编译命令（不包含任何优化开关）

|           |   |
|-----------|---|
| 对于 C++ 文件 | <code>g++.exe %s.cpp -o %s.exe -lm</code> |
| 对于 C 文件   | <code>gcc.exe %s.c -o %s.exe -lm</code>   |

## 注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中的函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 评测时可以使用万能头文件 `#include <bits/stdc++.h>`，默认支持 c++14。
4. 程序执行时堆栈空间限制与内存空间限制相同。
5. 提交的文件目录如下图所示，HB-0088 为考生准考证号，date、robot、tax 为题目规定的英文名称，目录中只包含源程序，不能包含其他任何文件。

|   |                 |
|---|-----------------|
|  HB-0088 | ----- 准考证号建立文件夹 |
|  date    | ----- date.cpp  |
|  robot   | ----- robot.cpp |
|  tax     | ----- tax.cpp   |

## 搭积木(box)

### 【题目描述】

小G喜欢搭积木。小G一共有 $n$ 块积木，并且积木只能竖着一块一块的摞，可以摞多列。小G的积木都是智能积木，第 $i$ 块积木有一个情绪值 $X_i$ 。当摞在该积木上面积木总数超过 $X_i$ 时， $i$ 号积木就会不高兴。小G情商这么高，肯定不希望有积木不高兴。但是他又希望每块积木都被用上，并且摞的积木列的总数最少。你能帮帮萌萌的小G吗？

### 【输入格式】

从文件 box.in 中读入数据。

输入文件第一行一个数字 $n$ ，含义如题目所述。

第2行一共 $n$ 个数，第 $i$ 个数为 $X_i$ ，含义如题目所述。

### 【输出格式】

输出到文件 box.out 中。

输出一个数字，表示最小的积木列数目。

### 【样例输入 1】

```
3
0 0 10
```

### 【样例输出 1】

```
2
```

### 【样例输入 2】

```
4
0 0 0 0
```

### 【样例输出 2】

```
4
```

### 【数据范围与约定】

30% 数据， $1 \leq n \leq 10$ 。

60% 数据， $1 \leq n \leq 100$ 。

80% 数据， $1 \leq n \leq 1000$ 。

100% 数据， $1 \leq n \leq 5000$ 。

对于所有数据点，都有  $X_i \leq n$ 。

## 城堡 (castle)

### 【题目描述】

小G 家有一座城堡。城堡里面有 $n$  个房间，每个房间上都写着一个数字 $p_i$ 。小G 拉着几个小伙伴在城堡里面玩耍，他们约定，如果某个人当前站在 $i$  房间里，下一步这个人就会去 $p_i$  房间，再下一步这个人去 $p_{p_i}$ 。

为了增加趣味性，小G 想重新书写每个房间的 $p_i$ ，以满足：

- 如果从编号1 到 $k$  中的某个房间开始，按照规则走，必须能够走到1 号房间。特别地，如果从1 号房间开始走，也要能够走回1 号房间（至少走一步，如果 $p_1 = 1$ ，从1 走到1 也算合法）。

- 如果从编号大于 $k$  的某个房间开始，按照规则走，一定不能走到1 号房间。

小G 想知道，有多少种书写 $p_i$  的方案，可以满足要求。

### 【输入格式】

从文件 castle.in 中读入数据。

输入文件一行两个数字  $n, k$ ，含义如题。

### 【输出格式】

输出到文件 castle.out 中。

输出一个整数，表示合法的方案数，答案对  $10^9 + 7$  取模。

### 【样例输入 1】

5 2

### 【样例输出 1】

54

### 【样例输入 2】

7 4

### 【样例输出 2】

1728

### 【数据范围与约定】

对于40% 的数据， $1 \leq n \leq 8$ 。

对于70% 的数据， $1 \leq n \leq 10^5$ 。

对于 100% 的数据， $1 \leq n \leq 10^{18}$ ， $1 \leq k \leq \min(8, n)$ 。

## 跳跃 (jump)

### 【题目描述】

美丽国有 $n$ 座小岛, 这些小岛排成了一条直线, 依次编号为 $0, 1, 2, 3 \dots n-1$ 。有的小岛上有金块, 有的小岛上没有。美丽国有一个巨人Bob, 有一天他决定从第0个岛开始, 一直往后跳, 看看自己能收集多少金块。当然, Bob不是没有目的的乱跳, Bob的跳跃要满足以下规则:

1. 首先Bob从0跳到 $d$ 岛屿;
2. 假如Bob上一次跳了 $L$ 步, 则Bob下一次可以选择跳 $L, L-1, L+1$ 步。

注意: 所跳的步数不能小于1, 即如果上一次跳跃的步数为1, 下一次跳跃的步数只能是1或者2。

在这样的规则下, Bob想知道, 他最多能拿到多少金块?

### 【输入格式】

从文件 `jump.in` 中读入数据。

输入第一行两个整数 $n, d$ , 分别表示有金块的岛屿的个数, 和Bob第一步要跳的步数。

接下来 $n$ 行, 每行一个整数 $id$ , 表示编号为 $id$ 的岛屿上有金块。

### 【输出格式】

输出到文件 `jump.out` 中。

输出仅一行, 表示Bob最多能收集多少金块。

### 【样例输入 1】

```
4 10
10
21
27
35
```

### 【样例输出 1】

```
3
```

### 【样例输入 2】

```
8 8
9
19
28
36
45
55
66
78
```

### 【样例输出 2】

6

### 【数据范围与约定】

对于30% 的数据,  $1 \leq n, d, id \leq 100$

对于70% 的数据,  $1 \leq n, d, id \leq 1000$

对于100% 的数据,  $1 \leq n, d, id \leq 30000$