# (tree.pas/c/cpp) 128M1s

## 【问题描述】

现在给出一棵二叉树,树上的非叶节点有两种,第一种是加节点,另一种是 max 节点。 加节点的值为左右节点的值之和,max 节点的值为左右节点的值中最大的哪一个。现在再 给你叶节点个数那么多个值,让你把他们填入对应的叶节点,使得根节点的值最大。用字符 'X'代表叶子节点,'A' 代表加节点,'B' 代表 max 节点。输入为这棵树的先序遍历。

## 【输入格式】

第一行一个仅含有'A','B','X'字符的字符串表示这棵树。

接下来一行,一个整数 T 代表叶节点个数。

接下来一行,有 T 个整数 V1···Vn, 为给定的值;

## 【输出格式】

输出一个整数表示根节点最大值。

## 【样例输入】

**AXBXX** 

3

823

【样例输出】

11

## 【数据规模及约定】

对于 30%的数据, 叶子节点个数≤10;

对于 60%的数据, 叶子节点个数≤19;

对于 100%的数据, 叶子节点个数≤2,000,0<=Vi<=100,000。

## 序列

(seq.pas/c/cpp) 256M1s

## 【问题描述】

有一个长度为 n 的非负整数序列 Ai ,每次操作可以选择相邻的两个数 Ai Ai+1,删去它们,然后在这个位置插入一个数 max(Ai,Ai+1),此次操作的代价定义为 max(Ai,Ai+1),求将这个序列长度变为 1 的最少代价。

## 【输入格式】

第一行为一个正整数 n. 表示序列的长度。

第二行有 n 个非负整数 Ai. 表示这个序列。

#### 【输出格式】

一行一个数,表示最少代价。

# 【样例输入】

3

123

【样例输出】

5

## 【数据规模】

对于 30%的数据, n<=10

对于 50%的数据, n<=5000 对于 80%的数据, n<=100000 对于 100%的数据, n<=1000000

# 工作计划 (havefun.pas/c/cpp) 256M1s

# 【问题描述】

工作使艾奇快乐。

勤奋的工作为国家直接贡献了 GDP, 艾奇认为只要对国家有利,即使牺牲自己生命也心甘情愿,绝不会因为自己可能受到祸害而躲开。

当艾奇无聊的时候,她就会去工作,然而并不是每次工作都是轻松而愉悦的。

当天艾奇又一次来到了学校, 等待着她的是一个有 n 行 m 列的巨大的矩阵和 q

个任务。对于每个任务,艾奇被要求交换这个矩阵中的两个子矩阵。每个任务,艾奇会获得六个正整数 x1,y1,x2,y2,h,w。x1,y1 代表了第一个矩阵左上角的行列位置(即在第 x1 行第 y1 列); x2,y2 代表了第二个矩阵左上角的行列位置,h,w 代表了这两个矩阵的高和宽(即行数和列数)。数据保证所有需要交换的矩阵互不相交或相邻。也就是说,没有任何一个元素同时属于这两个矩阵,也不存在某两个元素分别属于两个矩阵且相邻(共边)。

## 【输入格式】

输入文件名为 havefun.in。

第一行两个正整数 n,m,q 表示矩阵的高和宽(即行数和列数)和任务数。

接下来一行五个参数,将用于生成矩阵,详见备注。

接下来 q 行, 每行 6 个数, x1,y1,x2,y2,h,w,分别表示第一个矩阵的左上位置,

第二个矩阵的左上位置,这两个矩阵的高和宽(即行数和列数)。

## 【输出格式】

输出文件名为 havefun.out。

输出一个数、将用于校验你的矩阵、详见备注。

## 【样例数据】

见选手下发文件

### 【数据规模及约定】

数据编号	n,m	q	备注
1	100	100	
2	100	1000	
3	100	10000	
4	400	40000	
5	1000	100	
6	2000	1000	矩阵所有元素为0或1且满足
7	2000	1000	y1 mod 32 = 1
			y2 mod 32 = 1
			w mod 32 = 0
8	2000	1000	
9	2000	1000	
10	2000	1000	