

CSP2019-S 模拟题（第一试）

（请选手务必仔细阅读此页内容）

一．题目概况

中文题目名称	石子合并	寿司	反击
英文题目名称	stone	sushi	attack
可执行文件名	stone	sushi	attack
输入文件名	stone.in	sushi.in	attack.in
输出文件名	stone.out	sushi.out	attack.out
每个测试点时限	1 秒	1 秒	1 秒
内存上限	256MB	256MB	256MB
测试点数目	20	20	20
每个测试点分值	5	5	5
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统

二．提交源程序文件名

对于 Pascal 语言	stone.pas	sushi.pas	attack.pas
对于 C 语言	stone.c	sushi.c	attack.c
对于 C++ 语言	stone.cpp	sushi.cpp	attack.cpp

三．编译命令

对于 Pascal 语言	fpc stone.pas	fpc sushi.pas	fpc attack.pas
对于 C 语言	gcc -o stone stone.c -lm	gcc -o sushi sushi.c -lm	gcc -o attack Attack.c -lm
对于 C++ 语言	g++ -o stone stone.cpp -lm	g++ -o sushi sushi.cpp -lm	g++ -o attack attack.cpp -lm

四．注意事项：

1. 文件名（程序名和输入输出文件名）必须使用小写。
2. 选手提交以自己编号命名的文件夹，文件夹内包含 3 个源文件(.c, .cpp, .pas,)，并在文件夹下建立 3 个相应的子目录，并将 3 个对应的源程序分别放入对应的子文件夹中，所有名字必须使用小写；例如：

```
001          编号
|---stone
|          |---stone.cpp
|---sushi
|          |---sushi.cpp
|---attack
|          |---attack.cpp
|---stone.cpp
|---sushi.cpp
|---attack.cpp
```

3. C/C++中函数 `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`。
4. 题目简单，请认真对待，争取三位数。
5. 每道题源代码长度限制均为 `50KB`。
6. 每道题的数据都有一定梯度。请尽量优化算法，争取拿高分。
7. 编译时不打开任何优化选项。
8. 建议最后 10 分钟不要再编程，检查一下提交的文件夹中的代码是否符合要求，检查文件名，输入输出文件名，数据类型，数据精度，空间限制，赋初值等是否按试卷上的要求来做的，一定要杜绝一切的不小心的人为错误，显然这种错误是致命的。
9. 做题时，审题是关键，必须深入与全面，学过的知识与做过的题都是分析问题的有利武器；编写代码要细致，多写函数，便于调试，只有这样，才能达到你的期望。

必须杜绝答案错误！

1. 石子合并

(stone.pas/c/cpp)

【问题描述】

现在有 n 堆石子排成一列，其中第 i 堆石子有 A_i 个。每次将两堆石子合并成一堆，新堆的石子数是原来两堆石子数之和，合并的代价是原来两堆石子数之积。求合并 $n - 1$ 次将 n 堆石子合并成一堆的代价和最小值是多少。

【输入】

输入文件 stone.in

第一行输入一个正整数 n ;

第二行输入 n 个正整数，第 i 个正整数表示 A_i ，相邻两个数之间用空格隔开

【输出】

一个正整数，表示答案。

【输入输出样例 1】

stone.in	stone.out
2	4
2 2	

【输入输出样例 2】

stone.in	stone.out
4	101
2 3 7 5	

【输入输出样例 2 解释】

A_1 和 A_2 合并成新的一堆，石子数为 5，合并的代价为 6，然后这新的一堆与原来的 A_4 合并成新的一堆，石子数是 10，合并的代价为 25,最后这新的一堆与 A_3 合并，合并后的石子数为 17，合并的代价为 70，最后总的合并的最小代价为 $6+25+70=101$ 。

【数据说明】

测试点编号	n	其他约束
1、2	≤ 300	所有 $A_i < 10^6$ 都相同
3、4	≤ 2000	
5、6	≤ 300	
7、8	≤ 500	所有的 $A_i < 10^6$
9、10	≤ 1000	
11、12	≤ 2000	
13、14	≤ 30000	
15	≤ 4000000	
16	≤ 4000000	
17	≤ 4000000	
18	≤ 4000000	
19	≤ 4000000	
20	≤ 4000000	

2. 寿司

(sushi.pas/c/cpp)

【题目描述】

小 c 是一名 *oier*。最近，他发现他的数据结构好像学傻了。因为他在刷题时碰到了一道傻逼数据结构题，强行使用了平衡树来解决，卡着时间 AC 。为此，他被狠狠地嘲讽了一番。于是，小 c 找了大量的数据结构题来做。

昨天，小 c 正在吃寿司，突然发现许多盘寿司围成了一个圆圈，这些寿司中有红色的也有蓝色的。由于小 c 看交错的颜色非常不爽，想通过一些操作，使得所有的红色寿司形成了一块连续的区域，蓝色的寿司也形成了一块连续的区域。如果小 c 每次只可以交换相邻的两盘寿司，那么最少需要多少步才可以达到小 c 的要求呢？由于他做题做多了，脑袋已经有点不清醒了，于是这个问题就交给你了。

【输入】

输入文件 sushi.in

第一行一个数 T ，表示数据组数。

接下来 T 行，每行一行由 B 和 R 组成的字符串， B 表示蓝色， R 表示红色。第 i 个字符描述顺时针数第 i 盘寿司的颜色。注意，最后一盘寿司和第 1 盘寿司是相邻的。

【输出】

对于每组数据，输出一行表示最小的交换次数。

【输入输出样例】

sushi.in	sushi.out
1 BBRBBRBBRRR	5

【输入输出样例说明】

以下说明交换的步骤：

交换位置 2、位置 3 上的寿司；

交换位置 1、位置 2 上的寿司；

交换位置 6、位置 7 上的寿司；

交换位置 7、位置 8 上的寿司；

交换位置 8、位置 9 上的寿司；

【数据说明】

读入数据较大，请优化读入方式。

设 n 为字符串长度，则有：

测试点标号	$T \leq$	$n \leq$	测试点标号	$T \leq$	$n \leq$
1	1	10	11	5	50000
2	1	15	12	10	50000
3	1	20	13	10	100000
4	5	20	14	10	200000
5	2	2000	15	10	300000
6	2	4000	16	10	500000
7	2	5000	17	10	800000
8	3	5000	18	10	1000000
9	10	10000	19	10	1000000
10	5	20000	20	10	1000000

注：请注意优化常数

3、反击

(attack.pas/c/cpp)

【题目描述】

由于地球人和三体人的科技相差太过悬殊，尽管地球舰队顽强抗击，在三体人的高科技面前，地球军节节败退，形势危急。关键时刻，地球统帅 1j 决定发起一场重创三体舰队，挽救地球的自卫反击战。

通过三体人不会伪装的特性，1j 得知了三体人能源的来源——三星树。这是一棵能为三体舰队源源不断地提供能源的神树，树上的每个结点都有一个权值，表示这个结点可以为三体舰队能提供的能源值。现在 1j 召集了一支精锐部队，打算潜入三体舰队内部，并摧毁掉三星树。然而，三体人虽然不懂得伪装，但对于神树的保护却做得极为严密。出于安全考虑，1j 把这支部队分为了两支小队，两支小队可以分别选择三星树中的任何一个结点登陆（但不能从同一个位置），潜入到三星树中。并且，他们可以任意地沿着三星树的树边移动，每当他们到达一个三星树的结点，他们就可以不费吹灰之力地摧毁掉三星树这个结点。为了降低被发现的可能，两只小队都不能走回头路，而且他们走过的路径不能有任何公共结点，否则一大波人的行动的能量波动会引发三星树的报警装置，从而使这次任务失败。

他们可以选择在树上的任何位置传送出三星树，登上早已安排好的飞船迅速返回地球。1j 向两支小队的队长下的命令是，在保证不被发现的前提下，不惜一切代价，尽可能多的摧毁三星树的能源供应。所以两只小队的队长 Z 和 X 想知道，应该如何行动，他们才能够使摧毁的三星树的能源值总和最大。

【输入数据】

输入数据第一行仅包含一个数 n ，表示树的结点个数；

接下来一行包含 n 个数，第 i 个数 a_i 表示结点 i 的能源值为 a_i ；

第 3 至 $n+1$ 行，每行两个数 x, y ($1 \leq x, y \leq n$)，表示 x 和 y 之间存在一条树边。

【输出数据】

仅包含一个数，表示两支小队能够摧毁的能源值总和的最大值。

【输入输出样例】

attack.in	attack.out
8 1 2 99 4 5 3 2 1 1 2 1 4 1 6 1 3 4 5 6 7 6 8	115

【样例解释 1】

第一支小队从 3 登陆，经过 3-1-4-5 后，从 5 离开，总共摧毁能源值 109；
第二支小队从 7 登陆，经过 7-6-8 后，从 8 离开，总共摧毁能源值 6；
所以两支小队最多能摧毁的能源值为 115。

【输入输出样例 2】

attack.in	attack.out
9 1 2 3 4 5 6 7 8 9 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9	25

【样例解释 2】

第一支小队从 9 登陆，摧毁掉 9 后，直接从 9 离开，总共摧毁能源值 9；
第二支小队从 8 登陆后，经过 8-1-7 这一条路径后，从 7 离开，总共摧毁能源值 16；
所以两支小队最多能摧毁的能源值为 25。

【数据约定】

测试点编号	n<=	数据特点	测试点编号	n<=	数据特点	
1	10	保证数据随机	11	100000	保证每个点的 度数不超过 6	
2			12			
3			13			
4	1000		14		无	
5			15			
6			16			
7	100000		17			
8			18			
9			19			
10			20			

对于 100% 的数据，保证 a_i 均为小于等于 10^9 的正整数。