

Notice: 1:由于本OJ建立在Linux平台下，而许多题的数据在Windows下制作，请注意输入、输出语句及数据类型及范围，避免无谓的RE出现。 2:本站即将推出针对初学者的试题系统(与目前OJ是分开的，互不影响)，内容覆盖从语法入门到NOI的所有知识点，敬请关注。

2158: Crash 的旅行计划

Time Limit: 20 Sec Memory Limit: 259 MB

Submit: 59 Solved: 33

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

Description

过不了多久，Crash 就要迎来他朝思暮想的暑假。在这个暑假里，他计划着到火星上旅游。在火星上有N 个旅游景点，Crash 用1 至N 这N 个正整数对这些景点标号。旅游景点之间通过双向道路相连。由于火星的环境和地球有很大的差异，建立道路的成本也相对较高。为了节约成本，只有N-1 条道路连接着这些旅游景点，不过可以保证任何两个不同的旅游景点都通过路径相连。Crash 预先在互联网上查阅了这些景点的信息，根据网上的介绍，他对每个景点都有一个印象值，这个印象值为一个整数。在这个旅行中，他会选择一个景点作为旅行的开始，并沿着存在的道路到达其他景点游玩。为了使旅行不显得乏味，Crash 不会经过同一个景点超过一次。Crash 还给这次旅行定义了一个快乐指数，也就是他经过的所有景点的印象值之和。不过Crash 是个奇怪的小朋友，他对于景点的印象值会发生改变，并且他也没有决定好应该从哪个景点开始旅行。因此他希望你写一个程序帮他完成一个简单的任务：根据当前他对每个景点的印象值，计算从某个景点开始旅行所能获得的最大的快乐指数。

Input

输入的第一行包含一个字符和一个正整数N，字符为ABC 中的一个，用来表示这个测试数据的类型（详见下面的数据规模和约定）。第二行包含N 个用空格隔开的整数，第i 个整数表示Crash 对i 号景点的初始印象值。接着有N-1 行，每行两个正整数a、b($1 \leq a, b \leq N$)，表示从a 号景点到b 号景点有一条无向道路相连。最后是一些指令，指令只会是以下三种格式：1. Change u x ($1 \leq u \leq N$) 将u 号景

点的印象值修改为x。 2 . Query u ($1 \leq u \leq N$) 询问从u 号景点开始能获得的最大的快乐指数。 3 . Done 收到这个指令后，你的程序应该结束。

Output

对于每条Query 指令，输出对应的最大快乐指数。。

Sample Input

[样例输入1]

A 6

6 5 -4 3 -2 1

1 2

1 3

1 4

3 5

3 6

Query 3

Query 4

Change 6 10

Query 3

Change 2 -5

Query 3

Query 4

Done

[样例输入2]

B 5

5 -4 3 -2 1

1 2

2 3

3 4

4 5

Query 3

Change 5 10

Query 3

Query 2

Change 2 2

Query 3

Done

Sample Output

[样例输出1]

7

14

7

6

15

[样例输出2]

4

11

7

HINT

测试数据分为ABC 三类，对于所有的测试数据都满足：在任何时候一个景点印象值的绝对值不超过10000，并且输入的道路一定能满足题目描述的要求，即使得任意两个不同的景点都能通过路径相连。对于A 类数据（占20%的分数）满足：N 和指令的条数都不超过1000。对于B 类数据（占40%的分数）满足：N 和指令的条数都不超过100000，且输入的第i 条道路，连接着i 号景点和i+1 号景点（详见样例2）。对于C 类数据（占40%的分数）满足：N 和指令的条数都不超过100000，且任何一个景点到1 号景点需要通过的道路条数不超过40。【特别说明】由于数据大小限制为5MB，我只好对测试时的输入文件进行压缩处理。下面的函数可以将压缩的输入文件转化为原始输入文件。（函数从infile 中读入压缩的输入文件，将解压缩后的输入文件输出到outfile 中） C/C++版本：

```
void Uncompress(FILE *infile, FILE *outfile) { int N, M, L, now, A, B, Q, tmp, i; char type = getc(infile); fscanf(infile, "%d%d%d", &N, &M, &L); fscanf(infile, "%d%d%d%d", &now, &A, &B, &Q); fprintf(outfile, "%c %d\n", type, N); for (i = 1; i <= N; i++) { now = (now * A + B) % Q, tmp = now % 10000; now = (now * A + B) % Q; if (now * 2 < Q) tmp *= -1; if (i < N) fprintf(outfile, "%d ", tmp); else fprintf(outfile, "%d\n", tmp); } for (i = 1; i < N; i++) { now = (now * A + B) % Q; tmp = (i < L) ? i : L; fprintf(outfile, "%d %d\n", i - now % tmp, i + 1); } for (i = 1; i < M; i++) { now = (now * A + B) % Q; if (now * 3 < Q) { now = (now * A + B) % Q; fprintf(outfile, "Query %d\n", now % N + 1); } else { now = (now * A + B) % Q, tmp = now % 10000; now = (now * A + B) % Q; if (now * 2 < Q) tmp *= -1; now = (now * A + B) % Q; fprintf(outfile, "Change %d %d\n", now % N + 1, tmp); } } fprintf(outfile, "Done\n"); }
```

Pascal 版本：

```
procedure Uncompress(var infile, outfile : text); var N, M, L, now, A, B, Q, tmp, i : longint; ch : char; begin read(infile, ch, N, M, L, now, A, B, Q); writeln(outfile, ch, ' ', N); for i := 1 to N do begin now := (now * A + B) mod Q; tmp := now mod 10000; now := (now * A + B) mod Q; if now * 2 < Q then tmp := -tmp; if i < n then write(outfile, tmp, ' ') else writeln(outfile, tmp); end; for i := 1 to N - 1 do begin now := (now * A + B) mod Q; if i < L then tmp := i else tmp := L; writeln(outfile, i - now mod tmp, ' ', i + 1); end; for i := 1 to M - 1 do begin now := (now * A + B) mod Q; if now * 3 < Q then begin now := (now * A + B) mod Q; writeln(outfile, 'Query ', now mod N + 1); end else begin now := (now * A + B) mod Q; tmp := now mod 10000; now := (now * A + B) mod Q; if now * 2 < Q then tmp := -tmp; now := (now * A + B) mod Q; writeln(outfile, 'Change ', now mod N + 1, ' ', tmp); end; end; writeln(outfile, 'Done'); end;
```

下面给出一个具体的例子。travel_compressed.in 表示压缩的输入文

```
件 , travel.in 表示解压缩后的输入文件。 travel_compressed.in travel.in A 5 7 3
17627 543 14278 380043 travel.in A 5 -4664 7653 -3584 -210 5852 1 2 1 3 2 4 3
5 Change 5 -3724 Query 4 Change 3 -5628 Query 2 Change 5 569 Query 5
Done
```

Source

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

[HOME](#) [Back](#)

[한국어](#) [中文](#) [فارسی](#) [English](#) [ไทย](#)

版权所有 ©2008-2012 大视野在线测评 | 湘ICP备13009380号 | 站长统计

Based on opensource project hustoj.