标题：平方十位数

由 0~9 这 10 个数字不重复、不遗漏，可以组成很多 10 位数字。
这其中也有很多恰好是平方数（是某个数的平方）。

比如：1026753849，就是其中最小的一个平方数。

请你找出其中最大的一个平方数是多少?

注意：你需要提交的是一个 10 位数字，不要填写任何多余内容。

标题：生命游戏

康威生命游戏是英国数学家约翰·何顿·康威在 1970 年发明的细胞自动机。
这个游戏在一个无限大的 2D 网格上进行。

初始时，每个小方格中居住着一个活着或死了的细胞。
下一时刻每个细胞的状态都由它周围八个格子的细胞状态决定。

具体来说：

1. 当前细胞为存活状态时，当周围低于 2 个（不包含 2 个）存活细胞时， 该细胞变成死亡状态。(模拟生命数量稀少)
2. 当前细胞为存活状态时，当周围有 2 个或 3 个存活细胞时， 该细胞保持原样。
3. 当前细胞为存活状态时，当周围有 3 个以上的存活细胞时，该细胞变成死亡状态。(模拟生命数量过多)
4. 当前细胞为死亡状态时，当周围有 3 个存活细胞时，该细胞变成存活状态。 (模拟繁殖)

当前代所有细胞同时被以上规则处理后，可以得到下一代细胞图。按规则继续处理这一代的细胞图，可以得到再下一代的细胞图，周而复始。

例如假设初始是:(X 代表活细胞，.代表死细胞)
.....
.....
.XXX.
.....

下一代会变为:
.....
..X..
..X..
..X..

.....

康威生命游戏中会出现一些有趣的模式。例如稳定不变的模式：

....
.XX.
.XX.
....

还有会循环的模式：

```
......          ......          ......
.XX...          .XX...          .XX...
.XX...          .X....          .XX...
...XX.    -> ....X.    ->     ...XX.
...XX.          ...XX.          ...XX.
......          ......          ......
```

本题中我们要讨论的是一个非常特殊的模式，被称作"Gosper glider gun"：

```
........................................
........................X...............
......................X.X...............
............XX......XX............XX.
...........X...X....XX............XX.
.XX........X.....X...XX.................
.XX........X...X.XX....X.X..............
...........X.....X.......X..............
...........X...X........................
............XX..........................
........................................
```

假设以上初始状态是第 0 代，请问第 1000000000(十亿)代一共有多少活着的细胞?

注意：我们假定细胞机在无限的 2D 网格上推演，并非只有题目中画出的那点空间。
当然，对于遥远的位置，其初始状态一概为死细胞。

注意：需要提交的是一个整数，不要填写多余内容。

标题：表达式计算

虽然我们学了许久的程序设计，但对于简单的四则混合运算式，如果让我们完全白手起家地

编程来解析，还是有点棘手．

这里，我们简化一下问题，假设只有加法和乘法，并且没有括号来改变优先级．
再假设参加运算的都是正整数．

在这么多的限制条件下，表达式的解析似乎简单了许多．
下面的代码解决了这个问题．请仔细阅读源码，并填写划线部分缺少的代码．

```
#include <stdio.h>

int f3(const char* s, int begin, int end)
{
    int sum = 0;
    int i;
    for(i=begin; i<end; i++){
        if(s[i]==' ') continue;
        sum = sum * 10 + (s[i]-'0');
    }
    return sum;
}

int f2(const char* s, int begin, int end)
{
    int p = begin;
    int pro = 1;
    while(1){
        int p0 = p;
        while(p!=end && s[p]!='*') p++;
        pro *= ____________________________;  //填空
        if(p==end) break;
        p++;
    }
    printf("f2: pro=%d\n", pro);
    return pro;
}

int f(const char* s)
{
    int p = 0;
    int sum = 0;
    while(1){
        int p0 = p;
        while(s[p]!=0 && s[p]!='+') p++;
```

```
            sum += f2(s,p0,p);
            if(s[p]==0) break;
            p++;
        }

        return sum;
}

int main()
{
        int x = f("12+18+5*4*3+10");
        printf("%d\n", x);
        return 0;
}
```

注意：只填写划线处缺少的内容，不要填写已有的代码或符号，也不要填写任何解释说明文字等。

标题：填字母游戏

小明经常玩 LOL 游戏上瘾，一次他想挑战 K 大师，不料 K 大师说：
“我们先来玩个空格填字母的游戏，要是你不能赢我，就再别玩 LOL 了”。

K 大师在纸上画了一行 n 个格子，要小明和他交替往其中填入字母。

并且：

1. 轮到某人填的时候，只能在某个空格中填入 L 或 O
2. 谁先让字母组成了“LOL”的字样，谁获胜。
3. 如果所有格子都填满了，仍无法组成 LOL，则平局。

小明试验了几次都输了，他很惭愧，希望你能用计算机帮他解开这个谜。

本题的输入格式为：
第一行，数字 n（n<10），表示下面有 n 个初始局面。
接下来，n 行，每行一个串(长度<20)，表示开始的局面。
比如：“*******”，表示有 6 个空格。
“L****”， 表示左边是一个字母 L，它的右边是 4 个空格。

要求输出 n 个数字，表示对每个局面，如果小明先填，当 K 大师总是用最强着法的时候，小明的最好结果。
1 表示能赢
-1 表示必输

0 表示可以逼平

例如，
输入：
4
***
L**L
L**L***L
L*****L

则程序应该输出：
0
-1
1
1

资源约定：
峰值内存消耗 < 256M
CPU 消耗 < 1000ms

请严格按要求输出，不要画蛇添足地打印类似：“请您输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意: main 函数需要返回 0
注意：只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。
注意：所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。

标题：区间移位

数轴上有 n 个闭区间：D1,...,Dn。
其中区间 Di 用一对整数[ai, bi]来描述，满足 ai < bi。
已知这些区间的长度之和至少有 10000。
所以，通过适当的移动这些区间，你总可以使得他们的“并”覆盖[0, 10000]——也就是说[0, 10000]这个区间内的每一个点都落于至少一个区间内。
你希望找一个移动方法，使得位移差最大的那个区间的位移量最小。

具体来说，假设你将 Di 移动到[ai+ci, bi+ci]这个位置。你希望使得 maxi{|ci|} 最小。

【输入格式】
输入的第一行包含一个整数 n，表示区间的数量。
接下来有 n 行，每行 2 个整数 ai, bi，以一个空格分开，表示区间[ai, bi]。
保证区间的长度之和至少是 10000。

【输出格式】
输出一个数字，表示答案。如果答案是整数，只输出整数部分。如果答案不是整数，输出时四舍五入保留一位小数。

【样例输入】
2
10 5010
4980 9980

【样例输出】
20

【样例说明】
第一个区间往左移动 10；第二个区间往右移动 20。

【样例输入】
4
0 4000
3000 5000
5001 8000
7000 10000
【样例输出】
0.5
【样例说明】
第 2 个区间往右移 0.5；第 3 个区间往左移 0.5 即可。

【数据规模与约定】
对于 30%的评测用例，1 <= n <= 10；
对于 100%的评测用例，1 <= n <= 10000，0 <= ai < bi <= 10000。

资源约定：
峰值内存消耗 < 256M
CPU 消耗 < 1000ms

请严格按要求输出，不要画蛇添足地打印类似：“请您输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意: main 函数需要返回 0
注意: 只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。
注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。

标题：数组操作

给出一个长度为 n 的数组 {A}，由 1 到 n 标号， 你需要维护 m 个操作。
操作分为三种，输入格式为：

1 L R d，将数组中下标 L 到 R 的位置都加上 d，即对于 L<=i<=R，执行 A[i]=A[i]+d。

2 L1 R1 L2 R2，将数组中下标为 L1 到 R1 的位置，赋值成 L2 到 R2 的值，保证 R1-L1=R2-L2。
换句话说先对 0<=i<=R2-L2 执行 B[i]=A[L2+i]，再对 0<=i<=R1-L1 执行 A[L1+i]=B[i]，其中 {B} 为一个临时数组。

3 L R，求数组中下标 L 到 R 的位置的和，即求出 ∑_(i=L 到 R) A_i 。

输入格式:
从标准输入读入数据。
第一行一个整数 Case，表示测试点编号，其中 Case=0 表示该点为样例。
第二行包含两个整数 n,m。保证 1<=n,m<=10^5。
第三行包含 n 个整数 A_i，表示这个数组的初值。保证 0<=A_i<=10^5。
接下来 m 每行描述一个操作，格式如问题描述所示。
对于操作中提到每个数，满足 0<=d<=10^5，1<=L<=R<=n，1<=L1<=R1<=n，1<=L2<=R2<=n，R1-L1=R2-L2。

输出格式:
输出到标准输出。
对于每次 3 操作输出一行一个数，表示求和的结果。

样例输入:
```
0
5 6
1 2 3 4 5
2 1 3 3 5
3 3 5
1 2 4 2
3 3 5
```

2 1 3 3 5
3 1 5

样例输出:
14
18
29

测试点	n,m	其他约束
1,2	<=10^3	无
3,4	<=10^5	没有 2 操作
5,6,7	<=10^5	n 为偶数，且所有 2 操作满足 L1=1,R1=n/2 ,L2=n/2+1,R2=n
8,9,10	<=10^5	无

资源约定:
峰值内存消耗 < 2048M
CPU 消耗 < 2000ms

请严格按要求输出，不要画蛇添足地打印类似：“请您输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意: main 函数需要返回 0
注意: 只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。
注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。