## Solution

将问题转化为平面上的几何问题：

从坐标(0,0)开始，将白色方块看作向右走一个单位长度，将黑色方块看作向上走一个单位长度。

现我们以2W, 3B, 1W, 3B, 4W, 2B, 1W为例，得到如下图形

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

连结起点和终点，问题就可以转化为求这条直线与路径在整点上的交点数 -1，可以在O(N)的时间内出解。

#include <bits/stdc++.h>

using namespace std;

inline int read()

{

 int x = 0; char ch;

 while (!isdigit(ch = getchar()));

 do x = x \* 10 + ch - '0'; while(isdigit(ch = getchar()));

 return x;

}

#define N 100005

int test, n, x[N]; char t[N];

int main()

{

 freopen("silly.in", "r", stdin);

 freopen("silly.out", "w", stdout);

 test = read();

 while (test--)

 {

 n = read();

 int cw = 0, cb = 0;

 for (int i = 1; i <= n; ++i)

 {

 x[i] = read(); t[i] = getchar();

 (t[i] == 'W' ? cw : cb) += x[i];

 }

 if (!(cw && cb))//只有一种字符，字符个数即为份数

 { printf("%d\n", cw + cb); continue; }

 int d = \_\_gcd(cw, cb), tw = 0, tb = 0, ans = 0;

 cw /= d; cb /= d;

 for (int i = 1; i <= n; ++i)

 {

 if (t[i] == 'W')

 {

 if (tb % cb) { tw += x[i]; continue; }

 int t = tb / cb \* cw;

 if (tw < t && t <= tw + x[i]) ++ans;

 tw += x[i];

 }

 else

 {

 if (tw % cw) { tb += x[i]; continue; }

 int t = tw / cw \* cb;

 if (tb < t && t <= tb + x[i]) ++ans;

 tb += x[i];

 }

 }

 printf("%d\n", ans);

 }

 return 0;

}