

## 二维树状数组，矩阵操作及单点查询状态

```
#include <cstdio>
#include <stack>
#include <cmath>
#include <queue>
#include <string>
#include <queue>
#include <cstring>
#include <iostream>
#include <algorithm>

#define lid id<<1
#define rid id<<1|1
#define closein cin.tie(0)
#define scac(a) scanf("%c",&a)
#define scad(a) scanf("%d",&a)
#define print(a) printf("%d\n",a)
#define debug printf("hello world")
#define form(i,n,m) for(int i=n;i<m;i++)
#define mfor(i,n,m) for(int i=n;i>m;i--)
#define nfor(i,n,m) for(int i=n;i>=m;i--)
#define forn(i,n,m) for(int i=n;i<=m;i++)
#define scadd(a,b) scanf("%d%d",&a,&b)
#define memset0(a) memset(a,0,sizeof(a))
#define scaddd(a,b,c) scanf("%d%d%d",&a,&b,&c)
#define scadddd(a,b,c,d) scanf("%d%d%d%d",&a,&b,&c,&d)

#define INF 0x3f3f3f3f
#define maxn 1005
typedef long long ll;
using namespace std;
//-----AC(^-^)AC-----\\

int n;
int tree[maxn][maxn];

int lowbit(int x)
{
    return x&(-x);
}

void update(int x,int y)//更新子矩阵
{
    for(int i=x;i<=n;i+=lowbit(i))
    {
        for(int j=y;j<=n;j+=lowbit(j))
        {
            tree[i][j]++;
        }
    }
}
```

```

    }
}
}
int query(int x,int y)//查询指定点状态
{
    int ans=0;
    for(int i=x;i>0;i-=lowbit(i))
    {
        for(int j=y;j>0;j-=lowbit(j))
        {
            ans+=tree[i][j];
        }
    }
    return ans;
}
int main()
{
    int x;
    scanf("%d",&x);
    while(x--)
    {
        int t;
        memset(tree,0,sizeof(tree));
        scanf("%d",&t);
        for(int i=0;i<t;i++)
        {
            char op[2];
            int x1,x2,y1,y2;
            scanf("%s",op);
            if(op[0]=='C')
            {
                scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
                update(x1,y1);
                update(x2+1,y1);
                update(x1,y2+1);
                update(x2+1,y2+1);
            }
            else
            {
                int x1,y1;
                scanf("%d%d",&x1,&y1);
                printf("%d\n",query(x1,y1)%2);
            }
        }
    }
    return 0;
}

```