```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    ll z,n;
    cin >> z;
    for(int k=0;k<z;k++){
        cin >> n;
        ll nr = n << 1;
        ll M[nr][nr];
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                cin >> M[i][j];
                M[i+n][j] = M[i][j];
                M[i][j+n] = M[i][j];
                M[i+n][j+n] = M[i][j];
                if(i>0) M[i][j] += M[i-1][j];
                if(j>0) M[i][j] += M[i][j-1];
                if(i>0 && j>0) M[i][j] -= M[i-1][j-1];
            }
        }
        for(int i=0;i<nr;i++){
            for(int j=0;j<nr;j++){
                if((i>=n or j>=n)){
                    if(i>0){
                        M[i][j] += M[i-1][j];
                    }
                    if(j>0){
                        M[i][j] += M[i][j-1];
                    }
                    if(i>0 && j>0){
                        M[i][j] -= M[i-1][j-1];
                    }
                }
            }
        }
        ll ans = -1000*100*100;
        int xi,yi,xf,yf;
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                for(int x=i;x<i+n;x++){
                    for(int y=j;y<j+n;y++){
                        ll at = M[x][y];
                        if(i>0) at -= M[i-1][y];
                        if(j>0) at -= M[x][j-1];
                        if(i>0 and j>0) at += M[i-1][j-1];
                        ans = max(ans,at);
                        if(ans == at){
                            xi = i;
                            yi = j;
                            xf = x;
                            yf = y;
                        }
                    }
                }
            }
        }
        if(xi==0 and yi==0 and ((xf==nr-1 and yf==n-1) or (xf==n-1 and yf==nr-1)))
                ans -= M[n-1][n-1];
        cout << ans << endl;
        //cout << "COORD I:  " << xi << " " << yi << endl
            // << "COORD F:  " << xf << " " << yf << endl;
    }
}

```