MATRIX1:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
using namespace std;
const int oo = 105;
int A[oo][oo];
int main(){
	int test; cin >> test;
	while(test--){
		int n; cin >> n;
		for(int i=0; i<n; i++){
			for(int j=0; j<n; j++) cin >> A[i][j];
		}
		for(int i=0; i<n; i++){
			// neu la hang chan thi ta in tu 0->n
			if(i % 2 == 0){
				for(int j=0; j<n; j++) cout << A[i][j] << " ";
			}
			// le thi in tu n-1 -> 0
			else{
				for(int j=n-1; j>=0; j--) cout << A[i][j] << " ";
			}
		}
		cout << endl;
	}
}
```

MATRIX3:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int oo = 105;
int A[oo][oo];
int main(){
	cin.tie(0); ios_base::sync_with_stdio(false);
	int test; cin >> test;
	while(test--){
		int n, m; cin >> n >> m;
		for(int i=0; i<n; i++){
			for(int j=0; j<m; j++) cin >> A[i][j];
		}
		int d = 0, row = n, col = m;
		int count = 0; // chi cho in ra dung n*m lan
		while(d <= n/2 && d <= m/2){
			for(int i=d; i<col && count++ < n*m ; i++)
				cout << A[d][i] << " ";
			for(int i=d+1; i<row && count++ < n*m; i++)
```

```
                        cout << A[i][col-1] << " ";
                for(int i=col-2; i>=d && count++ < n*m; i--)
                        cout << A[row-1][i] << " ";
                for(int i=row-2; i>d && count++ < n*m; i--)
                        cout << A[i][d] << " ";
                d++; row--; col--;
        }
        cout << endl;
    }
}
//l=3 s=4
//1 2 3 4
//5 6 7 8
//9 10 11 12
```

MATRIX4:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
using namespace std;
const int oo = 105;
/*
    y tuong dung BFS va mang B[] de danh dau nhung vi tri da di qua
    duyet toan bo vector v den khi nao ko con diem nao co the di
*/
int A[oo][oo], B[oo][oo] = {0};
// tao 1 class Local de Local.x - hang thu i va Local.y - cot thu j
// co the dung pair<first, second> thay cho class Local
class Local{
    public:
        int x, y;
};
vector<Local> v; // vector chua nhung vi tri ma tai do A[local.x][local.y] = 1
// ham tim vi tri neu xung quanh co the di(A[i][j] = 1) thi push vao queue
void findLocal(int row, int col, queue<Local> &qt){
    for(int i=row-1; i<=row+1; i++){
        for(int j=col-1; j<=col+1; j++){
            if(A[i][j] && !B[i][j]){
                B[i][j] = 1;
                Local lc; lc.x = i; lc.y = j;
                qt.push(lc);
            }
        }
    }
}
// BFS
int BreadthFirstSearch(){
```

```
	queue<Local> q;
	int count = 0;
	for(int i=0; i<v.size(); i++){
		Local lcl = v[i];
		if(!B[lcl.x][lcl.y]){
			B[lcl.x][lcl.y] = 1;
			queue<Local> qt; qt.push(lcl);
			while(!qt.empty()){
				Local lt = qt.front(); qt.pop();
				int row = lt.x, col = lt.y;
				findLocal(row, col, qt);
			}
			count++;
		}
	}
	return count;
}
int main(){
	faster();
   int test; cin >> test;
   while(test--){
	v.clear();
	int n, m; cin >> n >> m;
	for(int i=1; i<=n; i++){
		for(int j=1; j<=m; j++){
			cin >> A[i][j]; B[i][j] = 0;
			// neu A[i][j] = 1 thi them vao vector v vi tri Local.i va Local.j
			if(A[i][j]){
				Local lp;
				lp.x = i; lp.y = j;
				v.push_back(lp);
				}
			}
		}
		int ans = BreadthFirstSearch();
		cout << ans << endl;
   }

}
```

MATRIX5:

```
// I'mAlone-CAN NGOC BINH
#include<iostream>
using namespace std;
const int oo = 105;
int A[oo][oo], n;
/*
	y tuong: tim max tong cac hang va cac cot
```

```
        luon co 1 phep bien doi sao cho tat cac cac hang = max nay
*/
// tong max hang
int maxSumRow(){
        int res = 0;
        for(int i=0; i<n; i++){
                int sum = 0;
                for(int j=0; j<n; j++) sum += A[i][j];
                res = max(sum, res);
        }
        return res;
}
// tong max cot
int maxSumColumn(){
        int res = 0;
        for(int i=0; i<n; i++){
                int sum = 0;
                for(int j=0; j<n; j++) sum += A[j][i];
                res = max(sum, res);
        }
        return res;
}
int main(){
   int test; cin >> test;
   while(test--){
      cin >> n;
      // tong cua cac mang A
      int sumArr = 0;
      for(int i=0; i<n; i++){
         for(int j=0; j<n; j++){
                cin >> A[i][j];
                sumArr += A[i][j];
                        }
      }
      int MAX = max(maxSumColumn(), maxSumRow());
      int ans = MAX*2*n - 2*sumArr;
      cout << ans/2 << endl;
   }
}
```

MATRIX6:

```
// I'mAlone-NGOC BINH
#include<bits/stdc++.h>
using namespace std;
#define ll long long
int main(){
        int test; cin >> test;
        while(test--){
```

```
		int n; cin >> n;
		int B[n+1][n+1];
		for(int i=0; i<n; i++){
			for(int j=0; j<n; j++){
				cin >> B[i][j];
				if(i == 0 || i == n-1 || j == 0 || j == n-1) cout << B[i][j] << " ";
				else cout << " ";
			}
			cout << endl;
		}
		cout << endl;
	}
}
```

MATRIX7:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
using namespace std;
const int oo = 105;
int A[oo][oo];
int main(){
	int test; cin >> test;
	while(test--){
		int n; cin >> n;
		for(int i=0; i<n; i++){
			for(int j=0; j<n; j++) cin >> A[i][j];
		}
		for(int i=0; i<n; i++){
			// neu la hang chan thi ta in tu 0->n
			if(i % 2 == 0){
				for(int j=0; j<n; j++) cout << A[i][j] << " ";
			}
			// le thi in tu n-1 -> 0
			else{
				for(int j=n-1; j>=0; j--) cout << A[i][j] << " ";
			}
		}
		cout << endl;
	}
}
```

MATRIX8:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define pb push_back
#define all(a) a.begin(), a.end()
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
using namespace std;
```

```
const int oo = 50;
int A[oo][oo], n;
/*
n = 2;
1  2  3  4  5  6  7  8
9  10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
cuon 1: 36 28 20 21 22 30 38 46 54 53 52 51 50 42 34 26 18 10 2 3 4 5 6 7 8 16 24 32 40
48 56 64
cuon 2: 29 37 45 44 43 35 27 19 11 12 13 14 15 23 31 39 47 55 63 62 61 60 59 58 57 49 41
33 25 17 9 1
*/
void solved(){
	/*
		ta tim cuon thu 2(COIL_A) truoc;
		sau do COIL_B[i] = 4*n*4*n - COIL_A[i] + 1;
	*/
	int SIZE = 4*n*4*n;
	vector<int> COIL_A, COIL_B;
	int col_start = 4*n, row_start = 4*n-2, col_end = 4*n-2, row_end = 4*n-4;
	int value = 1-4*n;
	while(COIL_A.size() < SIZE/2){
		for(int i=0; i<col_start; i++){
			value += 4*n;
			COIL_A.pb(value);
		}
		for(int i=0; i<row_start; i++){
			value++;
			COIL_A.pb(value);
		}
		for(int i=0; i<col_end; i++){
			value -= 4*n;
			COIL_A.pb(value);
		}
		for(int i=0; i<row_end; i++){
			value--;
			COIL_A.pb(value);
		}
		row_end -= 4; row_start -= 4;
		col_end -= 4; col_start -= 4;
	}
	reverse(all(COIL_A));
	for(auto ele: COIL_A) COIL_B.pb(SIZE - ele + 1);
```

```
	for(auto ele: COIL_B) cout << ele << " ";
	cout << endl;
	for(auto ele: COIL_A) cout << ele << " ";
}
int main(){
	faster();
	int test; cin >> test;
	while(test--){
		cin >> n;
		int res = 0;
		for(int i=0; i<4*n; i++){
			for(int j=0; j<4*n; j++) A[i][j] = ++res;
		}
		solved();
		cout << endl;
	}
}
```

MATRIX9:

```
// I'mAlone-CAN NGOC BINH
#include<iostream>
#include<queue>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
using namespace std;
const int oo = 2*1e3;
int A[oo][oo] = {0}, visit[oo][oo] = {0};
int n, m, x, y, z, t;
/*
	y tuong dung thuat toan tim kiem theo chieu rong BFS de duyet
	mang visit[] de danh dau da di
*/
int BFS(){
	int ans = 0;
	if(!A[x+1][y+1]) return -1;
	queue<pair<int, int> > q;
	// pair<first, second> first la hang thu i - second: cot thu j
	q.push(pair<int, int>(x+1,y+1));
	while(!q.empty()){
		int len = 0, SIZE = q.size();
		// ta chi lap dung SIZE lan (vi ta can tim so buoc di ngan nhat)
		while(len < SIZE){
			pair<int, int> local = q.front(); q.pop();
			int i = local.first, j = local.second;
			if(i == z+1 && j == t+1) return ans;
			// neu A[i][j] = 1(co the di) thi ta push vao queue va danh dau visit[i][j] =
1
			if(A[i+1][j] && !visit[i+1][j]){
				q.push(pair<int, int>(i+1, j));
```

```
                                visit[i+1][j] = 1;
                        }
                        if(A[i][j+1] && !visit[i][j+1]){
                                q.push(pair<int, int>(i, j+1));
                                visit[i][j+1] = 1;
                        }
                        if(A[i-1][j] && !visit[i-1][j]){
                                q.push(pair<int, int>(i-1, j));
                                visit[i-1][j] = 1;
                        }
                        if(A[i][j-1] && !visit[i][j-1]){
                                q.push(pair<int, int>(i, j-1));
                                visit[i][j-1] = 1;
                        }
                        len++;
                }
                ans++;
        }
        return -1;
}
int main(){
        faster();
        int test; cin >> test;
        while(test--){
                cin >> n >> m >> x >> y >> z >> t;
                for(int i=1; i<=n; i++){
                        for(int j=1; j<=m; j++){
                                cin >> A[i][j]; visit[i][j] = 0;
                        }
                }
                cout << BFS() << endl;
        }
}
```

MATRIX10:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int oo = 105;
int A[oo][oo], B[oo][oo];
// dung 1 mang B de mang A ko thay doi
int main(){
        int test; cin >> test;
        while(test--){
                int n, m; cin >> n >> m;
                for(int i=0; i<n; i++){
                        for(int j=0; j<m; j++){
```

```
                        cin >> A[i][j];
                        B[i][j] = A[i][j];
                    }
                }
                for(int i=0; i<n; i++){
                    for(int j=0; j<m; j++){
                        if(A[i][j] == 1){
                            // cho cot j = 1
                            for(int k=0; k<n; k++) B[k][j] = 1;
                            // cho hang i bang 1
                            for(int k=0; k<m; k++) B[i][k] = 1;
                        }
                    }
                }
                for(int i=0; i<n; i++){
                    for(int j=0; j<m; j++) cout << B[i][j] << " ";
                    cout << endl;
                }
                cout << endl;
        }
}
```

MATRIX11:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
using namespace std;
int A[20][20];
// dem so cot toan 1 tu hang l->r
int maxRectangle(int l, int r, int m){
        int res = 0;
        for(int i=1; i<=m; i++){
            bool isFull_1 = false;
            for(int j=l; j<=r; j++){
                if(!A[j][i]){
                    isFull_1 = true;
                    break;
                }
            }
            if(!isFull_1) res++;
        }
        return res;
}
int main(){
        faster();
        int test; cin >> test;
        while(test--){
            int n, m; cin >> n >> m;
```

```
		for(int i=1; i<=n; i++){
			for(int j=1; j<=m; j++) cin >> A[i][j];
		}
		int ans = 0;
		for(int i=1; i<=n; i++){
			for(int j=i; j<=n; j++){
				// dien tich cua hcn
				ans = max(ans, maxRectangle(i, j, m) * (j-i+1));
			}
		}
		cout << ans << endl;
	}
}
```

MATRIX13:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
using namespace std;
const int mod = 1e9+7;
const int oo = 505;
/*
	y tuong su dung ki thuat deque MIN/MAX tren doan tinh tien(co the su dung STACK)
	tai hang thu i ta dung quy hoach dong de tinh H[i][j]
	H[i][j]: la do dai day 1 lien tiep tinh xet o thu hang i cot j
	vi H[i][j] chi phu thuoc vao H[i-1][j] nen ta co the toi uu bo nho bang mang 1 chieu H[i]
	ky thuat deque:
		de bai yeu cau voi moi vi tri i tim vi tri left va right
		sao cho left < i < right va A[i] = min(A[left->right])
		cho mang A[] = {1,5,6,4,3,3,4,6,2}
		voi thuat toan trau bo thi mat O(n^2)
		voi deque khi xet 1 phan tu A[i] thi ta co the biet ngay dc vi tri left va right
		(cac b co the search gg ve deque)
*/
int A[oo][oo], H[oo] = {0};
int main(){
	faster();
   int test; cin >> test;
   while(test--){
	int n, m; cin >> n >> m;
	for(int i=1; i<=n; i++){
		for(int j=1; j<=m; j++) cin >> A[i][j];
		}
		int ans = 0;
		for(int i=1; i<=n; i++){
			// quy hoach dong tim H[i]
			for(int j=1; j<=m; j++) H[j] = (A[i][j])? H[j] + 1 : 0;
```

```
                        int L[oo], R[oo], D[oo], P[oo];
                        // L[i], R[i]: lan luot la 2 vi tri xa nhat ma H[i] = min(H[L[i] -> R[i]])
                        // tuc la H[i] <= H[k] voi moi k thuoc khoang [L[i], R[i]]
                        // cac ban co the dung STACK(code minh de ben duoi) thay cho cho
mang D va P
                        int top = 0;
                        D[0] = 0;
                        for(int j=1; j<=m; j++){
                                while(top > 0 && H[D[top]] >= H[j]) top--;
                                L[j] = D[top] + 1;
                                D[++top] = j;
                        }
                        P[0] = m+1; top = 0;
                        for(int j=m; j>=1; j--){
                                while(top > 0 && H[P[top]] >= H[j]) top--;
                                R[j] = P[top] - 1;
                                P[++top] = j;
                        }
                        for(int j=1; j<=m; j++) ans = max(ans, (R[j] - L[j] + 1) * H[j]);
                }
                // code su dung stack minh de ben duoi
                cout << ans << endl;
   }
}
/*
#include<iostream>
#include<algorithm>
#include<cmath>
#include<vector>
#include<stack>
#include<queue>
#define ll long long
using namespace std;
const int mod = 1e9+7;
const int oo = 505;
int A[oo][oo], H[oo] = {0};
int main(){
        cin.tie(0); ios_base::sync_with_stdio(false);
   int test; cin >> test;
   while(test--){
        int n, m; cin >> n >> m;
        for(int i=1; i<=n; i++){
                for(int j=1; j<=m; j++) cin >> A[i][j];
                }
                int ans = 0;
                for(int i=1; i<=n; i++){
                        for(int j=1; j<=m; j++) H[j] = (A[i][j])? H[j] + 1 : 0;
                        int L[oo], R[oo];
```

```
                        stack<int> stc1, stc2;
                        stc1.push(0);
                        for(int j=1; j<=m; j++){
                                while(stc1.size() > 1 && H[stc1.top()] >= H[j]) stc1.pop();
                                L[j] = stc1.top() + 1;
                                stc1.push(j);
                        }
                        stc2.push(m+1);
                        for(int j=m; j>=1; j--){
                                while(stc2.size() > 1 && H[stc2.top()] >= H[j]) stc2.pop();
                                R[j] = stc2.top() - 1;
                                stc2.push(j);
                        }
                        for(int j=1; j<=m; j++) ans = max(ans, (R[j] - L[j] + 1) * H[j]);
                }
                cout << ans << endl;
   }

}
*/

MATRIX14:
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
const int MAX = 505;
using namespace std;
int A[MAX][MAX], dp[MAX][MAX] = {0};
/*
        su dung quy hoach dong
        dp[i][j] la hinh vuong 1-1 lon nhat xet den vi tri hang i cot j
        o hang i cot j la o goc phai ben duoi cua hinh vuong
*/
int main(){
        faster();
   int test; cin >> test;
   while(test--){
        int n, m; cin >> n >> m;
        for(int i=1; i<=n; i++){
                for(int j=1; j<=m; j++) cin >> A[i][j];
                }
                int ans = 0;
        for(int i=1; i<=n; i++){
                for(int j=1; j<=m; j++){
                        if(A[i][j]){
                                /*
```

```
                                        1 1 1
                                        0 1 1
                                        trong truong hop nay se giai thich vi sao la min ma ko
phai max
                                        neu la max thi dp[3][3] = 3 (sai)
                                */
                                dp[i][j] = min(dp[i-1][j], min(dp[i-1][j-1], dp[i][j-1]));
                                dp[i][j] += 1;
                                }
                                else dp[i][j] = 0;
                                ans = max(ans, dp[i][j]);
                        }
                }
                cout << ans << endl;
   }
}
```

MATRIX15:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
using namespace std;
const int mod = 1e9+7;
const int oo = 505;
int A[oo][oo], B[oo][oo] = {0}, n, m;
vector<pair<int, int> > v;
/*
        y tuong:
                dung thuat toan tim kiem theo chieu rong BFS
                va mang B de danh dau cac row va col da di
                (B[][] = 1 : da di / B[][] = 0: chua di)
                ta di tu cac phan tu O vong ben ngoai vao ben trong
                vi vay neu o thu i, j nao = 'C' ta cho A[i][j] = 1;
                va them no vao vector v
                - sau do duyet toan bo vector v
*/
// tim duong di xung quanh
void haveRoad(int row, int col, queue<pair<int, int> > &q){
        if(A[row-1][col] && !B[row-1][col]) q.push(pair<int, int>(row-1, col));
        if(A[row][col-1] && !B[row][col-1]) q.push(pair<int, int>(row, col-1));
        if(A[row+1][col] && !B[row+1][col]) q.push(pair<int, int>(row+1, col));
        if(A[row][col+1] && !B[row][col+1]) q.push(pair<int, int>(row, col+1));
}
void BFS(){
        for(int i=0; i<v.size(); i++){
                if(!B[v[i].first][v[i].second]){
                        queue<pair<int, int> > q;
```

```
                        if(!B[v[i].first][v[i].second])       q.push(v[i]);
                        while(!q.empty()){
                                pair<int, int> par = q.front(); q.pop();
                                B[par.first][par.second] = 1;
                                haveRoad(par.first, par.second, q);
                        }
                }
        }
}
int main(){
        faster();
   int test; cin >> test;
   while(test--){
        cin >> n >> m;
        v.clear();
                for(int i=1; i<=n; i++){
                        for(int j=1; j<=m; j++){
                                B[i][j] = 0;
                                char c; cin >> c;
                                if(c == 'O'){
                                        // neu c == 'O' ta cho A[i][j] = 1 tuc la co the di
                                        A[i][j] = 1;
                                        // ta xet tu vong ngoai vao nen chi lay nhung vi tri tren
                                        if(i == 1 || j == 1 || i == n || j == m){
                                                pair<int, int> p = pair<int, int>(i, j);
                                                v.push_back(p);
                                        }
                                }
                                else A[i][j] = 0;
                        }
                }
                BFS();
                // cac phan tu B[i][j] = 1 thi thay the = 'O'
                // else thay the = 'X'
                for(int i=1; i<=n; i++){
                        for(int j=1; j<=m; j++){
                                if(B[i][j]) cout << 'O' << " ";
                                else cout << 'X' << " ";
                        }
                        cout << endl;
                }
                cout << endl;
   }

}
```

MATRIX16:

```
// I'mAlone-CAN NGOC BINH
```

```
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
using namespace std;
const int oo = 30;
pair<int, int> PAIR[oo][oo];
// pair[i][j] -first: do dai X dai nhat tinh theo row cua matran tai o thu hang i cot j
// - second: do dai canh X dai nhat tinh theo column cua matran tai o thu hang i cot j
/*
X O X X X                   (1,1) (0,0) (1,1) (2,1) (3,1)
X X X X X                   (1,2) (2,1) (3,2) (4,2) (5,2)
X X O X O       PAIR->      (1,3) (2,2) (0,0) (1,3) (0,0)
X X X X X                   (1,4) (2,3) (3,1) (4,4) (5,1)
X X X O O                   (1,5) (2,4) (3,2) (0,0) (0,0)
*/
int A[oo][oo], n;
// set mang Pair chuan bi tinh do dai canh hinh vuong
void setPair(){
	for(int i=1; i<=n; i++){
		for(int j=1; j<=n; j++){
			if(A[i][j]){
				PAIR[i][j].first = PAIR[i][j-1].first + 1;
				PAIR[i][j].second = PAIR[i-1][j].second + 1;
			}
			else PAIR[i][j] = pair<int, int>(0, 0);
		}
	}
}
int solved(){
	faster();
	// xet tung hang- xet do dai canh hinh vuong
	int ans = 0;
	for(int i=1; i<=n; i++){
		int MAX = n-i+1;
		// xet do dai canh hinh vuong j->k
		for(int j=1; j<=n; j++){
			for(int k=j; k<=n; k++){
				if(!A[i][k] || k-j+1 > MAX) break;
				else{
					// tinh 4 canh cua hinh vuong
					int row1 = PAIR[i][k].first - PAIR[i][j].first + 1;
					if(row1 <= ans) continue;
					pair<int, int> p1 = PAIR[i+row1-1][j];
					pair<int, int> p2 = PAIR[i+row1-1][k];
					int row2 = p2.first - p1.first + 1;
					int col1 = p1.second - PAIR[i][j].second + 1;
					int col2 = p2.second - PAIR[i][k].second + 1;
```

```
                                        if(row1 == row2 && row1 == col1 && row1 == col2) ans
= max(ans, row1);
                                    }
                                }
                            }
                        }
                        return ans;
}
int main(){
                        cin.tie(0); ios_base::sync_with_stdio(false);
    int test; cin >> test;
    while(test--){
                        cin >> n;
                        // set PAIR ve (0,0)
                        for(int i=0; i<=n; i++){
                                    for(int j=0; j<=n; j++){
                                                PAIR[i][j].first = 0;
                                                PAIR[i][j].second = 0;
                                                }
                                    }
                                    for(int i=1; i<=n; i++){
                                                for(int j=1; j<=n; j++){
                                                            char c; cin >> c;
                                                            A[i][j] = (c == 'X')? 1 : 0;
                                                }
                                    }
                                    setPair();
                                    cout << solved() << endl;
    }
}

MATRIX17:
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define ll long long
using namespace std;
const int oo = 100;
/*
                        y tuong: su dung thuat toan quay lui(backTracking)
                                                de duyet tat ca cac cach di chuyen den A[n-1][n-1]
*/
ll A[oo][oo], s, n, k, ans;
void backTrack(int i, int j){
                        s += A[i][j];
                        if(i == n-1 && j == n-1){
                                    if(s == k) ans++;
                        }
```

```
	// neu s van con < k thi ta con duyet
	else if(s <= k){
		if(i < n-1) backTrack(i+1, j);
		if(j < n-1) backTrack(i, j+1);
	}
	// tra lai s luc chua cong A[i][j] cho thang dang sau no dung
	s -= A[i][j];
}
int main(){
	faster();
	int test; cin >> test;
	while(test--){
		cin >> n >> k;
		s = 0; ans = 0;
		for(int i=0; i<n; i++){
			for(int j=0; j<n; j++) cin >> A[i][j];
		}
		backTrack(0,0);
		cout << ans << endl;
	}
}
```

MATRIX19:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
using namespace std;
const int oo = 105;
int A[oo][oo], n, m;
/*
	y tuong bai nay la dung 1 bool move lam dieu huong
	00 01 02 03
	10 11 12 13
	20 21 22 23
	30 31 32 33
	increase_i di xuong(tang i)-move = false (03->12->21->30)
	increase_j di len(tang j)-move = true (31->22>13)
	dung index_i va index_j de luu tru gia tri hang i cot j
*/
// di xuong
void increase_i(int &index_i, int &index_j){
	int limit = index_i;
	cout << A[index_i][index_j] << " ";
	while(index_j > limit){
		index_i++; index_j--;
		cout << A[index_i][index_j] << " ";
	}
}
```

```
// di len
void increase_j(int &index_i, int &index_j){
	int limit = index_j;
	cout << A[index_i][index_j] << " ";
	while(index_i > limit){
		index_i--; index_j++;
		cout << A[index_i][index_j] << " ";
	}
}
int main(){
	faster();
	int test; cin >> test;
	while(test--){
		cin >> n >> m;
		for(int i=0; i<n; i++){
			for(int j=0; j<n; j++) cin >> A[i][j];
		}
		int index_i = 0, index_j = 0;
		bool move = true; // true: di len / false: di xuong
		while(index_i < n && index_j < m){
			if(!move){
				increase_i(index_i, index_j);
				if(index_i == n-1) index_j++;
				else index_i++;
				move = true; // doi huong di len
			}
			else{
				increase_j(index_i, index_j);
				if(index_j == m-1) index_i++;
				else index_j++;
				move = false; // doi huong di xuong
			}
		}
		cout << endl;
	}
}
```

MATRIX20:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
#define pb push_back
#define ll long long
using namespace std;
const int oo = 105;
int A[oo][oo];
/*
	y tuong giong nhu bai in ma tran xoan oc nhung khac la
```

```
	ta push vao vector sau do moi in ra v[k-1]
*/
int main(){
	faster();
	int test; cin >> test;
	while(test--){
		vector<int> v;
		int n, m, k; cin >> n >> m >> k;
		for(int i=0; i<n; i++){
			for(int j=0; j<m; j++) cin >> A[i][j];
		}
		int d = 0, l = n, s = m;
		int count = 0;
		while(d <= n/2 && d <= m/2){
			for(int i=d; i<s; i++)
				v.pb(A[d][i]);
			for(int i=d+1; i<l; i++)
				v.pb(A[i][s-1]);
			for(int i=s-2; i>=d; i--)
				v.pb(A[l-1][i]);
			for(int i=l-2; i>d; i--)
				v.pb(A[i][d]);
			d++; l--; s--;
		}
		cout << v[k-1] << endl;
	}
}
```

MATRIX21:

```
// I'mAlone-CAN NGOC BINH
#include<bits/stdc++.h>
#define faster() cin.tie(0); ios_base::sync_with_stdio(false);
const int oo = 250;
#define ll long long
using namespace std;
ll A[oo][oo], dp[oo][oo] = {0}, n, m;
/*
	bai nay su dung quy hoach dong
	bai nay nen goi dp[i][j][k] la tong lon nhat cua day xet tu vi tri j->k trong hang i
	tuy nhien dp[i][j][k] chi phu thuoc vao dp[i-1][j][k] nen ta co the thay bang
	mang 2 chieu dp[i][j] tong lon nhat xet vi tri i->j trong 1 hang
*/
ll maxRectangle(){
	ll ans = 0;
	// xet tai hang thu i tu vi tri j->k
	for(int i=1; i<=n; i++){
		for(int j=1; j<=m; j++){
```

```
                        ll sum = 0;
                        for(int k=j; k<=m; k++){
                                sum += A[i][k];
                                // ta co the viet dp[j][k] = max(dp[j][k] + sum, sum)
                                dp[j][k] = (dp[j][k] >= 0)? dp[j][k] + sum : sum;
                                ans = max(ans, dp[j][k]);
                        }
                }
        }
        return ans;
}
int main(){
        faster();
   int test; cin >> test;
   while(test--){
        cin >> n >> m;
        for(int i=1; i<=n; i++){
                for(int j=1; j<=m; j++){
                        cin >> A[i][j];
                        }
                }
                // reset dp[] = 0
                for(int i=0; i<oo; i++){
                        for(int j=0; j<oo; j++) dp[i][j] = 0;
                }
                cout << maxRectangle() << endl;
   }
}
```