```cpp
//
//   There are N rabbits, numbered 1,2, ... N.
//   For each i,j (1 <= i,j <= N), the compatibility of
//   Rabbit i and j is described by an integer a[i][j]
//     a[i][i] = 0
//     a[i][j] = a[j][i]
//
//   Taro is dividing the N rabbits into some number of groups.
//   Here, each rabbit must belong to exactly one group.
//   After grouping, for each i and j (1 <= i < j <= N),
//   Taro earns a[i][j] points if Rabbit i and j belong to the same group.
//
//   Find Taro's maximum possible total score.
//
//      Time Complexity: O(2^N * N^2 + 2^(2N))
//

#include <bits/stdc++.h>
#define ll long long

using namespace std;

const int mxn = (1 << 16) + 10;
ll dp[mxn];

int main() {
    for (int i = 0; i < mxn; i++) {
        dp[i] = 0;
    }

    int n;
    cin >> n;

    vector<vector<int>> a(n, vector<int>(n, 0));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }

    for (int mask = 0; mask < (1 << n); mask++) {
        ll score = 0;
        for (int bit1 = 0; bit1 < n; bit1++) {
            for (int bit2 = bit1 + 1; bit2 < n; bit2++) {
                if (mask & (1 << bit1) && mask & (1 << bit2)) {
                    score += a[bit1][bit2];
                }
            }
        }

        dp[mask] = max(dp[mask], score);
    }

    for (int mask1 = 0; mask1 < (1 << n); mask1++) {
        for (int mask2 = mask1 + 1; mask2 < (1 << n); mask2++) {
            if ((mask1 & mask2) == 0) {
                dp[mask1 | mask2] = max(dp[mask1 | mask2], dp[mask1] + dp[mask2]);
            }
        }
    }

    cout << dp[(1 << n) - 1] << endl;
    return 0;
}
```