```cpp
#include <iostream>
#include <vector>

using namespace std;
typedef vector<int> vec;
typedef vector<vec> vec2;
typedef vector<vec2> vec3;

const int MEMO_INIT = -1;
vec v;
vec3 memo;

int f(int i, int j, int t, int m, int k) {
        int turn = t % m;
        // Base case, i = j
        if (i == j) {
                if (turn == k) return v[i];
                else return 0;
        }

        // Check memo
        if (memo[i][j][turn == k] != MEMO_INIT) return memo[i][j][turn == k];

        // Reccurence
        int result = MEMO_INIT;
        if (turn == k) {
                int take_first = v[i] + f(i+1, j, t+1, m, k);
                int take_last = v[j] + f(i, j-1, t+1, m, k);
                result = max(take_first, take_last);
        } else {
                int take_first = f(i+1, j, t+1, m, k);
                int take_last = f(i, j-1, t+1, m, k);
                result = min(take_first, take_last);
        }

        // Save in memo (just for fun ya know, meybe we will need it again)
        memo[i][j][turn == k] = result;
        return result;
}

void testcase() {
        int n, m, k; cin >> n >> m >> k;
        v = vec(n);
        for(int i = 0; i < n; i++) {
                cin >> v[i];
        }
        memo = vec3(n, vec2(n, vec(2, MEMO_INIT)));

        cout << f(0, n - 1, 0, m, k) << endl;
}

int main() {
        ios::sync_with_stdio(false);
        int t; cin >> t;
        while (t--) testcase();
        return 0;
}
```