

## Code submission: 100%

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

using namespace std;
typedef vector<int> vint;

int n, m, k;

int profit(vint &coins, vector<vint> &memo, int i, int j) {
    if (j < i) return 0;
    if (memo[i][j] != -1) return memo[i][j];

    int p1 = profit(coins, memo, i + 1, j);
    int p2 = profit(coins, memo, i, j - 1);
    bool myturn = (n - (j - i + 1)) % m == k;

    if (myturn) return memo[i][j] = max(coins[i] + p1, coins[j] + p2);
    else return memo[i][j] = min(p1, p2);
}

void solve() {
    cin >> n >> m >> k;
    vint coins(n);
    vector<vint> memo(n, vint(n, -1));
    for (int i = 0; i < n; i++) cin >> coins[i];
    cout << profit(coins, memo, 0, n - 1) << endl;
}

int main() {
    ios_base::sync_with_stdio(false);
    int t; cin >> t;
    for (int i = 0; i < t; i++) solve();
}
```