dominoStomp.cpp

```
#include <cstdio>
#include <algorithm>
using namespace std;

const int Inf = 2000000017;
const int Maxn = 1005;
const int Maxm = 3;
const int Maxk = 1005;

int n, k;
int B[Maxn][Maxm];
int dp[2][Maxm][1 << Maxm][Maxk];
int cur = 0;

int Get(int c, int mask, int tk)
{
        int cr = cur;
        if (c == Maxm) { cr = !cur; c = 0; }
        return dp[cr][c][mask][tk];
}

int main()
{
        scanf("%d %d", &n, &k);
        for (int i = 0; i < n; i++)
                for (int j = 0; j < Maxm; j++)
                        scanf("%d", &B[i][j]);
        dp[cur][0][0][k] = 0;
        for (int tk = 0; tk < k; tk++) dp[cur][0][0][tk] = Inf;
        for (int r = n - 1; r >= 0; r--) {
                cur = !cur;
                for (int c = Maxm - 1; c >= 0; c--)
                        for (int mask = 0; mask < 1 << Maxm; mask++)
                                for (int tk = 0; tk <= k; tk++) {
                                        int res = -Inf, cand;
                                        if (!(mask & 1 << c) && r + 1 < n && tk < k) {
                                                cand = Get(c + 1, mask | 1 << c, tk + 1);
                                                if (cand != Inf) res = max(res, cand + B[r][c] + B[
r + 1][c]);
                                        }
                                        if (c + 1 < Maxm && !(mask & 1 << c) && !(mask & 1 << c + 1
) && tk < k) {
                                                cand = Get(c + 2, mask, tk + 1);
                                                if (cand != Inf) res = max(res, cand + B[r][c] + B[
r][c + 1]);
                                        }
                                        cand = (mask & 1 << c)? Get(c + 1, mask ^ 1 << c, tk): Get(
c + 1, mask, tk);
                                        if (cand != Inf) res = max(res, cand);
                                        if (res == -Inf) res = Inf;
                                        dp[cur][c][mask][tk] = res;
                                }
        }
        printf("%d\n", dp[cur][0][0][0]);
        return 0;
}
```