```cpp
//
// Taro's summer vacation starts tomorrow, and he has decided to make plans for it now.
// The vacation consists of N days. For each i (1≤i≤N),
// Taro will choose one of the following activities and do it on the i-th day:
//   A: Swim in the sea. Gain a[i] points of happiness.
//   B: Catch bugs in the mountains. Gain b[i] points of happiness.
//   C: Do homework at home. Gain c[i] points of happiness.
// As Taro gets bored easily, he cannot do the same activities for two or more consecutive days.
//
// Find the maximum possible total points of happiness that Taro gains.
//
//  Time Complexity: O(N)
//

#include <bits/stdc++.h>

using namespace std;

int main() {
    int n;
    cin >> n;

    vector<vector<int>> happy(n, vector<int>(3));
    for (int i = 0; i < n; i++) {
        for (int k = 0; k < 3; k++) {
            scanf("%d", &happy[i][k]);
        }
    }

    vector<vector<int>> dp(n, vector<int>(3, 0));

    for (int k = 0; k < 3; k++) {
        dp[0][k] = happy[0][k];
    }

    for (int i = 1; i < n; i++) {
        for (int k = 0; k < 3; k++) {
            int ans = 0;

            for (int l = 0; l < 3; l++) {
                if (l != k) ans = max(ans, dp[i-1][l]);
            }

            dp[i][k] = happy[i][k] + ans;
        }
    }

    int ans = 0;
    for (int i = 0; i < 3; i++) {
        ans = max(ans, dp[n-1][i]);
    }

    cout << ans << endl;
    return 0;
}
```