

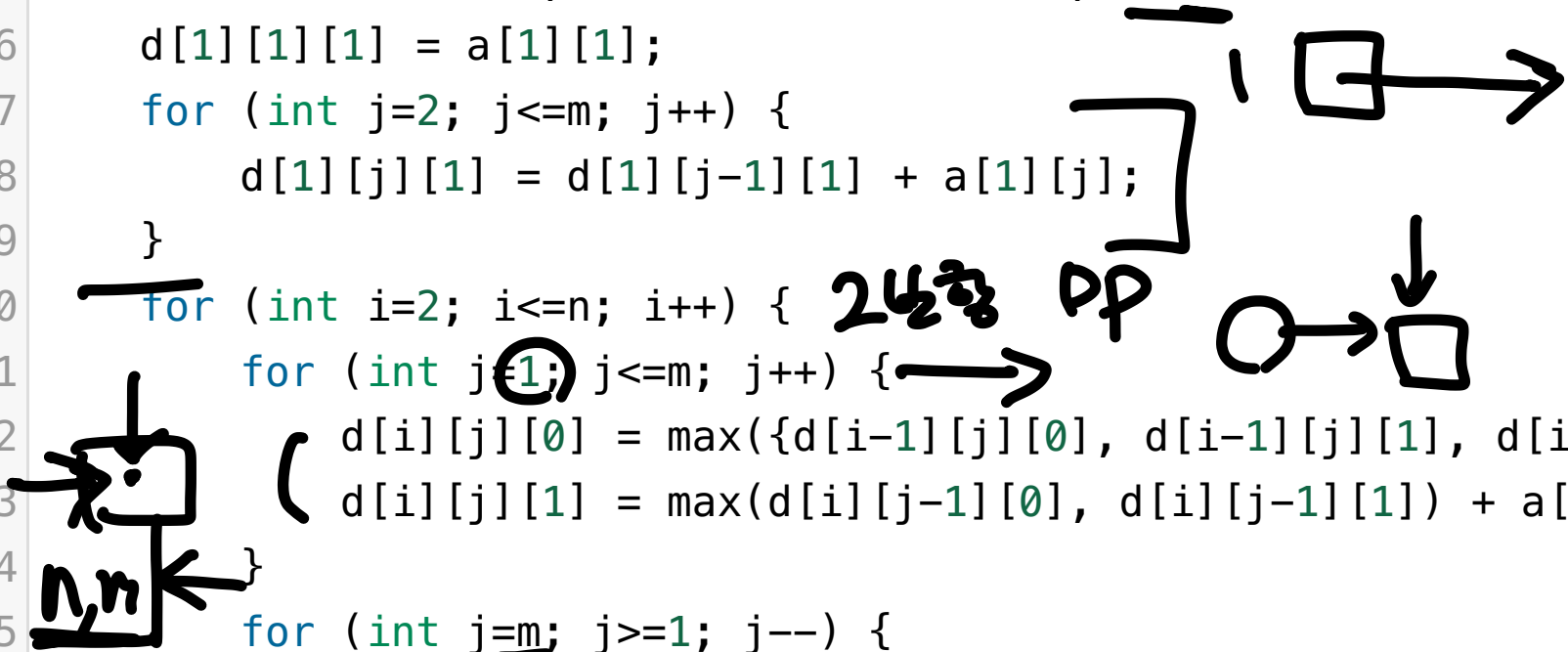
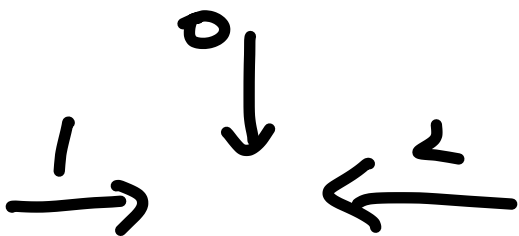
다미 나믹 프로그래밍 3 (민승)

소스코드

최백준 choi@startlink.io

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4 int a[1002][1002];
5 int d[1002][1002][3];
6 const int inf = 1000000000;
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    for (int i=1; i<=n; i++) {
11        for (int j=1; j<=m; j++) {
12            cin >> a[i][j];
13        }
14    }
15    fill(&d[0][0][0], &d[1001][1001][2]+1, -inf);
16    d[1][1][1] = a[1][1];
17    for (int j=2; j<=m; j++) {
18        d[1][j][1] = d[1][j-1][1] + a[1][j];
19    }
20    for (int i=2; i<=n; i++) {
21        for (int j=1; j<=m; j++) {
22            d[i][j][0] = max({d[i-1][j][0], d[i-1][j][1], d[i-1][j][2]}) + a[i][j];
23            d[i][j][1] = max(d[i][j-1][0], d[i][j-1][1]) + a[i][j];
24        }
25        for (int j=m; j>=1; j--) {
26            d[i][j][2] = max(d[i][j+1][0], d[i][j+1][2]) + a[i][j];
27        }
28    }
29    cout << max({d[n][m][0], d[n][m][1], d[n][m][2]}) << '\n';
30    return 0;
31 }
32
```



결과	메모리	시간	코드 길이
맞았습니다!!	17676 KB	228 ms	870 B

C++14

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 long long d[11][101][101][101];
5 long long c[11][11];
6 long long go(int n, int r, int g, int b) {
7     if (r < 0 || g < 0 || b < 0) {
8         return 0;
9     }
10    if (n == 0) {
11        return 1;
12    }
13    if (d[n][r][g][b] != -1) {
14        return d[n][r][g][b];
15    }
16    d[n][r][g][b] = 0;
17    d[n][r][g][b] += go(n-1, r-n, g, b);
18    d[n][r][g][b] += go(n-1, r, g-n, b);
19    d[n][r][g][b] += go(n-1, r, g, b-n);
20    if (n%2 == 0) {
21        d[n][r][g][b] += go(n-1, r-n/2, g-n/2, b)*c[n][n/2];
22        d[n][r][g][b] += go(n-1, r-n/2, g, b-n/2)*c[n][n/2];
23        d[n][r][g][b] += go(n-1, r, g-n/2, b-n/2)*c[n][n/2];
24    }
25    if (n%3 == 0) {
26        d[n][r][g][b] += go(n-1, r-n/3, g-n/3, b-n/3)*c[n][n/3]*c[n-n/3][n/3];
27    }
28    return d[n][r][g][b];
29 }
30 int main() {
31     int n,r,g,b;
32     cin >> n >> r >> g >> b;
33     c[0][0] = 1;
34     for (int i=1; i<=10; i++) {
35         c[i][0] = c[i][i] = 1;
36         for (int j=1; j<i; j++) {
37             c[i][j] = c[i-1][j-1] + c[i-1][j];
38         }
39     }
40     memset(d, -1, sizeof(d));
41     cout << go(n, r, g, b) << '\n';
42     return 0;
43 }
44
```

결과

메모리

시간

코드 길이

맞았습니다!!

90528 KB

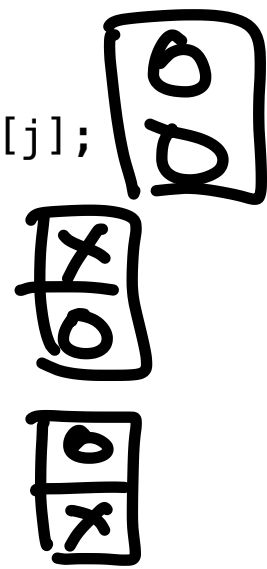
48 ms

1128 B

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int d[401][401][401];
6 int main() {
7     ios_base::sync_with_stdio(false);
8     int n;
9     cin >> n;
10    vector<int> a, b;
11    a.push_back(0);
12    b.push_back(0);
13    for (int i=0; i<n; i++) {
14        int temp;
15        cin >> temp;
16        if (temp != 0) {
17            a.push_back(temp);
18        }
19    }
20    for (int i=0; i<n; i++) {
21        int temp;
22        cin >> temp;
23        if (temp != 0) {
24            b.push_back(temp);
25        }
26    }
27    for (int k=1; k<=n; k++) {
28        for (int i=1; i<a.size(); i++) {
29            for (int j=1; j<b.size(); j++) {
30                if (i > k) continue;
31                if (j > k) continue;
32                int cur = d[k-1][i-1][j-1] + a[i]*b[j];
33                if (j-1 >= 0 && k-1 >= i) {
34                    cur = max(cur, d[k-1][i][j-1]);
35                }
36                if (i-1 >= 0 && k-1 >= j) {
37                    cur = max(cur, d[k-1][i-1][j]);
38                }
39                d[k][i][j] = cur;
40            }
41        }
42    }
43    cout << d[n][((int)a.size()-1)][((int)b.size()-1)] << '\n';
44    return 0;
45 }
```

위 a
아 b

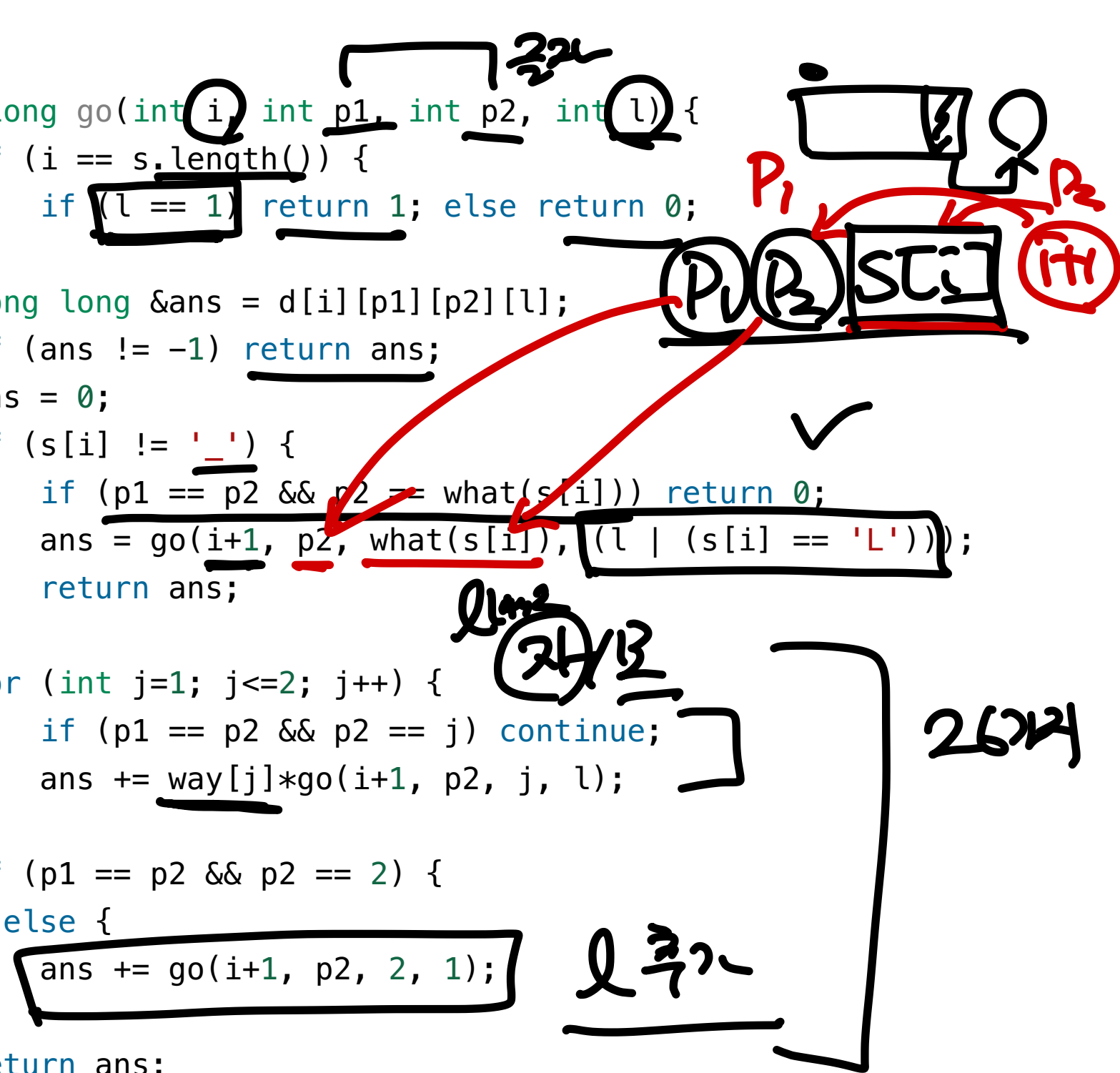


C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <string>
4 using namespace std;
5 string s;
6 long long d[111][3][3][2];
7 long long way[3] = {0, 5, 20};
8 // 1: mo, 2: ja
9 int what(char x) {
10     if (x == 'A' || x == 'E' || x == 'I' || x == 'O' || x == 'U') {
11         return 1;
12     } else {
13         return 2;
14     }
15 }
16 long long go(int i, int p1, int p2, int l) {
17     if (i == s.length()) {
18         if (l == 1) return 1; else return 0;
19     }
20     long long &ans = d[i][p1][p2][l];
21     if (ans != -1) return ans;
22     ans = 0;
23     if (s[i] != '_') {
24         if (p1 == p2 && p2 == what(s[i])) return 0;
25         ans = go(i+1, p2, what(s[i]), (l | (s[i] == 'L')));
26         return ans;
27     }
28     for (int j=1; j<=2; j++) {
29         if (p1 == p2 && p2 == j) continue;
30         ans += way[j]*go(i+1, p2, j, l);
31     }
32     if (p1 == p2 && p2 == 2) {
33     } else {
34         ans += go(i+1, p2, 2, 1);
35     }
36     return ans;
37 }
38 int main() {
39     cin >> s;
40     memset(d, -1, sizeof(d));
41     cout << go(0, 0, 0, 0) << '\n';
42     return 0;
43 }
44
```

what(x): x가 모음 1
 26 L 자음 2

way: 1 모음 계속
 2 자음 계속



끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.