

01背包:

```

1  #include<iostream>
2  #include<cstring>
3  using namespace std;
4
5  const int N = 1010;
6  int n, V, v[N], w[N];
7  int dp[N][N];
8  int main() {
9      cin >> n >> V;
10     for (int i = 1; i <= n; i++) {
11         cin >> v[i] >> w[i];
12     }
13
14     for (int i = 1; i <= n; i++) {
15         for (int j = 0; j <= V; j++) {
16             dp[i][j] = dp[i - 1][j];
17             if (j >= v[i])
18                 dp[i][j] = max(dp[i][j], dp[i][j - v[i]] + w[i]);
19         }
20     }
21     cout << dp[n][V] << endl;
22
23     memset(dp, 0, sizeof dp); // 将 dp 数组全部初始化为 0
24     for (int j = 1; j <= V; j++) dp[0][j] = -1; // 初始化第一行
25     for (int i = 1; i <= n; i++) {
26         for (int j = 0; j <= V; j++) {
27             dp[i][j] = dp[i - 1][j];
28             if (j >= v[i] && dp[i][j - v[i]] != -1)
29                 dp[i][j] = max(dp[i][j], w[i] + dp[i][j - v[i]]);
30         }
31     }
32
33     if (dp[n][V] == -1) {
34         cout << 0 << endl; // 表示无法装入有效的物品组合
35     } else {
36         cout << dp[n][V] << endl; // 输出最大价值
37     }
38     return 0;

```

