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

using namespace std;

class Node {
public:
    int value;
    int weight;
    Node(int value, int weight) : weight(weight), value(value) {};
};

class NodeResult {
public:
    int total_value;
    string total_digit;
    NodeResult() : total_value(0) {};
};

int main(int argc, char** argv) {

    ifstream file(argv[1]);

    vector<Node> nodes;

    int count = 0;
    int weight_r, value_r;
    while (file >> value_r >> weight_r) {
        nodes.push_back(Node(value_r, weight_r));
        count++;
    }

    NodeResult table[7][101];

    for (int i = 1; i <= count; i++) {
        int weight_t = nodes[i-1].weight;
        int value_t = nodes[i-1].value;
        for (int j = weight_t; j <= 100; j++) {
            if ((value_t + table[i-1][j-weight_t].total_value) > table[i-1]
[j].total_value) {
                table[i][j].total_value = value_t + table[i-1][j-weight_t].total_value;
                table[i][j].total_digit = table[i-1][j-weight_t].total_digit +
to_string(i);
            } else {
                table[i][j].total_value = table[i-1][j].total_value;
                table[i][j].total_digit = table[i-1][j].total_digit;
            }
        }
    }

    cout << "First let's see the diagram for this knapsack problem:" << endl;
    cout << endl;

    cout << "    ";
    for (int i = 0; i <= 6; i++) {
        if (i < 10) cout << i << "  ";
        else cout << i << " ";
    }

    cout << endl;

    for (int j = 0; j <= 100; j++) {
        if (j < 10) {
            cout << j << "   ";
        } else if (10 <= j && j < 100) {
            cout << j << "  ";
        } else {
            cout << j << " ";
        }

        for (int i = 0; i <= 6; i++) {
            if (table[i][j].total_value < 10) cout << table[i][j].total_value << "  ";
            else cout << table[i][j].total_value << " ";
        }
        cout << endl;
    }

    cout << endl;
    cout << "The final result is:" << endl;
    cout << endl;

    cout << "value is " << table[6][100].total_value << ", task is " << table[6]
[100].total_digit;

    return 0;
}
```

```
First let's see the diagram for this knapsack problem:

    0   1   2   3   4   5   6
0   0   0   0   0   0   0   0
1   0   0   0   0   0   0   0
2   0   0   0   0   0   0   0
3   0   0   0   0   0   0   0
4   0   0   0   0   0   0   0
5   0   0   0   0   0   0   2
6   0   0   0   0   0   0   2
7   0   0   0   0   0   0   2
8   0   0   0   0   0   0   2
9   0   0   0   0   0   0   2
10  0   0   0   0   0   10  10
11  0   0   0   0   0   10  10
12  0   0   0   0   0   10  10
13  0   0   0   0   0   10  10
14  0   0   0   0   0   10  10
15  0   0   0   0   0   10  12
16  0   0   0   0   0   10  12
17  0   0   0   0   0   10  12
18  0   0   0   0   0   10  12
19  0   0   0   0   0   10  12
20  0   0   0   0   4   10  12
21  0   0   0   0   4   10  12
22  0   0   0   0   4   10  12
23  0   0   0   0   4   10  12
24  0   0   0   0   4   10  12
25  0   0   0   0   4   10  12
26  0   0   0   0   4   10  12
27  0   0   0   0   4   10  12
28  0   0   0   0   4   10  12
29  0   0   0   0   4   10  12
30  0   0   0   0   4   14  14
31  0   0   0   0   4   14  14
32  0   0   0   0   4   14  14
33  0   0   0   0   4   14  14
34  0   0   0   0   4   14  14
35  0   0   0   0   4   14  16
36  0   0   0   0   4   14  16
37  0   0   0   0   4   14  16
38  0   0   0   0   4   14  16
39  0   0   0   0   4   14  16
40  0   0   0   0   4   14  16
41  0   0   0   0   4   14  16
42  0   0   0   0   4   14  16
43  0   0   0   0   4   14  16
44  0   0   0   0   4   14  16
45  0   0   0   18  18  18  18
46  0   0   0   18  18  18  18
47  0   0   0   18  18  18  18
48  0   0   0   18  18  18  18
49  0   0   0   18  18  18  18
50  0   0   35  35  35  35  35
51  0   0   35  35  35  35  35
52  0   0   35  35  35  35  35
```

```
53  0  0  35 35 35 35 35
54  0  0  35 35 35 35 35
55  0  0  35 35 35 35 37
56  0  0  35 35 35 35 37
57  0  0  35 35 35 35 37
58  0  0  35 35 35 35 37
59  0  0  35 35 35 35 37
60  0  0  35 35 35 45 45
61  0  0  35 35 35 45 45
62  0  0  35 35 35 45 45
63  0  0  35 35 35 45 45
64  0  0  35 35 35 45 45
65  0  0  35 35 35 45 47
66  0  0  35 35 35 45 47
67  0  0  35 35 35 45 47
68  0  0  35 35 35 45 47
69  0  0  35 35 35 45 47
70  0  0  35 35 39 45 47
71  0  0  35 35 39 45 47
72  0  0  35 35 39 45 47
73  0  0  35 35 39 45 47
74  0  0  35 35 39 45 47
75  0  0  35 35 39 45 47
76  0  0  35 35 39 45 47
77  0  0  35 35 39 45 47
78  0  0  35 35 39 45 47
79  0  0  35 35 39 45 47
80  0  0  35 35 39 49 49
81  0  0  35 35 39 49 49
82  0  0  35 35 39 49 49
83  0  0  35 35 39 49 49
84  0  0  35 35 39 49 49
85  0  0  35 35 39 49 51
86  0  0  35 35 39 49 51
87  0  0  35 35 39 49 51
88  0  0  35 35 39 49 51
89  0  0  35 35 39 49 51
90  0  0  35 35 39 49 51
91  0  0  35 35 39 49 51
92  0  0  35 35 39 49 51
93  0  0  35 35 39 49 51
94  0  0  35 35 39 49 51
95  0  0  35 53 53 53 53
96  0  0  35 53 53 53 53
97  0  0  35 53 53 53 53
98  0  0  35 53 53 53 53
99  0  0  35 53 53 53 53
100 0  40 40 53 53 53 55

The final result is:

value is 55, task is 236
```