## Input:

```
#include<bits/stdc++.h>
using namespace std;

struct item{
    int weight;
    int profit;
};

int find_max_profit(item items[], int capacity, int n){
    char ch;
    cout << "Do you want to view whch items are picked(y/n): ";
    cin >> ch;

    int result = 0;

    int dp[n+1][capacity+1];

    // setting row1 and column1 as zero
    for(int i=0; i<capacity+1; i++) dp[0][i] = 0;
    for(int i=0; i<n+1; i++) dp[i][0] = 0;
    cout << "\nDP table\n";
    for(int i=1; i<n+1; i++){
        for(int j=1; j<capacity+1; j++){
            if(items[i-1].weight > j){
                dp[i][j] = dp[i-1][j];
            }else{
                dp[i][j] = max(items[i-1].profit+dp[i-1][j-items[i-1].weight], dp[i-1][j]);
            }
            cout << dp[i][j] << " ";
        }
        cout <<  endl;
    }

    if(ch=='y'){
        int i=n, j=capacity;
        while(i>0){
            if(dp[i][j] == dp[i-1][j]){
                i--;
            }else{
                cout<<"\nItem picked: "<<i << " Weight-"<<items[i-1].weight << " Profit-"<<items[i-1].profit;
                j = j-items[i-1].weight;
                i--;
            }
        }
    }

    return dp[n][capacity];
}

int main(){
    int n, capacity;
    cout << "Enter the count of items: ";
    cin >> n;
    cout << "Enter capacity of bag: ";
    cin >> capacity;

    item items[n];

    cout << "Enter the items weight: ";
    int w;
    for(int i=0; i<n; i++){
        cin >> w;
        items[i].weight = w;
    }
    cout << "Enter the items profit: ";
    int p;
    for(int i=0; i<n; i++){
        cin >> p;
        items[i].profit = p;
    }

    int result = find_max_profit(items, capacity, n);
    cout << endl << endl << "Maximum profit: " <<result <<endl;

    return 0;
}
```

## **Output:**

```
@somesh4545 ➜ /workspaces/TE-Labs/DAA (main) $ g++ knapsack01.cpp && ./a.out
Enter the count of items: 4
Enter capacity of bag: 8
Enter the items weight: 2 3 4 5
Enter the items profit: 3 4 5 6
Do you want to view whch items are picked(y/n): y

DP table
0 3 3 3 3 3 3 3
0 3 4 4 7 7 7 7
0 3 4 5 7 8 9 9
0 3 4 5 7 8 9 10

Item picked: 4 Weight-5 Profit-6
Item picked: 2 Weight-3 Profit-4

Maximum profit: 10

@somesh4545 ➜ /workspaces/TE-Labs/DAA (main) $ g++ knapsack01.cpp && ./a.out
Enter the count of items: 5
Enter capacity of bag: 25
Enter the items weight: 5 10 15 8 1
Enter the items profit: 15 20 30 40 10
Do you want to view whch items are picked(y/n): y

DP table
0 0 0 0 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
0 0 0 0 15 15 15 15 15 20 20 20 20 20 35 35 35 35 35 35 35 35 35 35 35
0 0 0 0 15 15 15 15 15 20 20 20 20 20 35 35 35 35 35 45 45 45 45 45 50
0 0 0 0 15 15 15 40 40 40 40 40 55 55 55 55 55 60 60 60 60 60 75 75 75
10 10 10 10 15 25 25 40 50 50 50 50 55 65 65 65 65 65 70 70 70 70 75 85 85

Item picked: 5 Weight-1 Profit-10
Item picked: 4 Weight-8 Profit-40
Item picked: 2 Weight-10 Profit-20
Item picked: 1 Weight-5 Profit-15

Maximum profit: 85
```