2. Knapsack Problem

```
#include <iostream>

using namespace std;

  int max(int a, int b) {

        return (a > b) ? a : b;

}

  int main() {

        int n, W;

        cout << "Enter number of items: ";

        cin >> n;

        int wt[n], val[n];

        cout << "Enter weights:\n";

        for(int i = 0; i < n; i++)

                cin >> wt[i];

        cout << "Enter values:\n";

        for(int i = 0; i < n; i++)

                cin >> val[i];

      cout << "Enter capacity of knapsack: ";

        cin >> W;

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

        for(int i = 0; i <= n; i++) {

                for(int w = 0; w <= W; w++) {

                        if(i == 0 || w == 0)

                                dp[i][w] = 0;

                        else if(wt[i - 1] <= w)
```

```
                dp[i][w] = max(val[i - 1] + dp[i - 1][w - wt[i - 1]],
                               dp[i - 1][w]);
            else
                dp[i][w] = dp[i - 1][w];
        }
    }
    cout << "Maximum profit: " << dp[n][W];
    return 0;
}
```

Output

```
Enter number of items: 2
Enter weights:
2 3
Enter values:
10 20
Enter capacity of knapsack: 2
Maximum profit: 10

=== Code Execution Successful ===
```