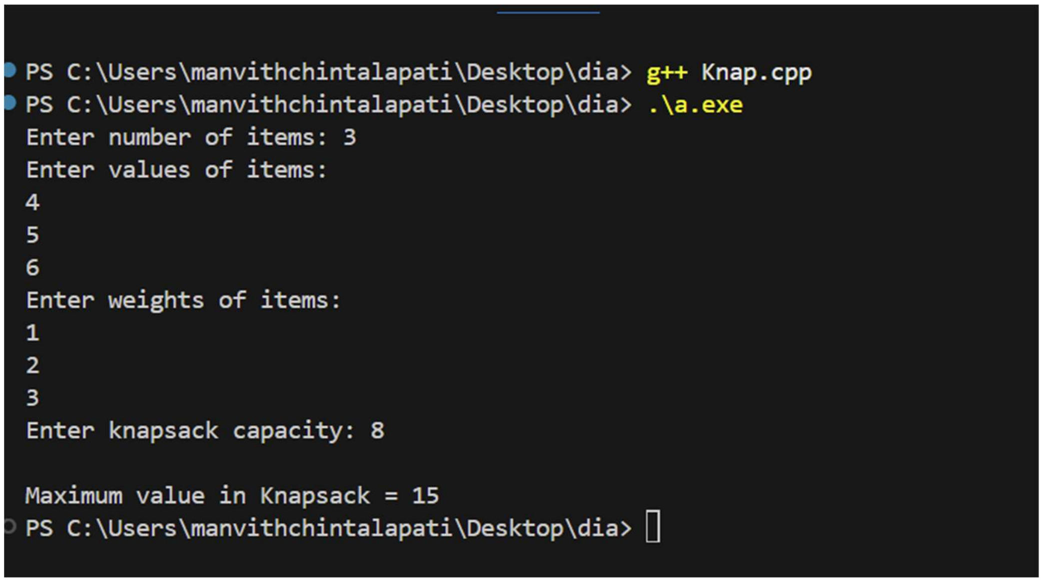
Dynamic Programming: Implement 0/1 Knapsack problem

```
#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;


int knapsack(int W, vector<int> &wt, vector<int> &val, int n) {

  vector<vector<int>> dp(n + 1, vector<int>(W + 1, 0));


  for (int i = 1; i <= n; i++) {

    for (int w = 1; w <= W; w++) {

      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];

    }

  }




  return dp[n][W];

}


int main() {

  int n, W;

  cout << "Enter number of items: ";

  cin >> n;
```

```
    vector<int> value(n), weight(n);

    cout << "Enter values of items:\n";
    for (int i = 0; i < n; i++)
        cin >> value[i];

    cout << "Enter weights of items:\n";
    for (int i = 0; i < n; i++)
        cin >> weight[i];

    cout << "Enter knapsack capacity: ";
    cin >> W;

    int maxValue = knapsack(W, weight, value, n);
    cout << "\nMaximum value in Knapsack = " << maxValue << endl;

    return 0;
}
```



```
PS C:\Users\manvithchintalapati\Desktop\dia> g++ Knap.cpp
PS C:\Users\manvithchintalapati\Desktop\dia> .\a.exe
Enter number of items: 3
Enter values of items:
4
5
6
Enter weights of items:
1
2
3
Enter knapsack capacity: 8

Maximum value in Knapsack = 15
PS C:\Users\manvithchintalapati\Desktop\dia> 
```