Name:-Dikesh Ganboi
Roll NO:- A36

```cpp
#include <iostream>
#include <vector>

using namespace std;

// Utility function to find maximum of two integers
int max(int a, int b) {
    return (a > b) ? a : b;
}

// Function to solve knapsack problem
int knapSack(int W, int wt[], int val[], int n) {
    // Initialize a 2D vector to store results of subproblems
    vector<vector<int>> K(n + 1, vector<int>(W + 1));

    // Build K[][] in bottom-up manner
    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= W; w++) {
            if (i == 0 || w == 0)
                K[i][w] = 0;
            else if (wt[i - 1] <= w)
                K[i][w] = max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i -
1][w]);
            else
                K[i][w] = K[i - 1][w];
        }
    }

    // K[n][W] contains the maximum value that can be obtained
    return K[n][W];
}

int main() {
    int val[] = {70, 20, 50}; // Values of items
    int wt[] = {11, 12, 13};   // Weights of items
    int W = 30;                 // Maximum weight capacity of knapsack
    int n = sizeof(val) / sizeof(val[0]); // Number of items

    // Print the maximum profit achieved with the knapsack
    cout << "Maximum Profit achieved with this knapsack: " << knapSack(W, wt,
val, n);
```

```
    return 0;
}
```

OUTPUT:-

```
PS C:\Users\HP\Desktop\DAA EXperiment> cd "c:\Users\HP\Desktop\DAA
EXperiment\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o
tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }

Maximum Profit achieved with this knapsack: 120
```