## CHANDIGARH UNIVERSITY

## UNIVERSITY INSTITUTE OF NGINEERING

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



|  |  |
| --- | --- |
| **Submitted By: Submitted To:**  Vivek Kumar(21BCS8129) Neha Dutta(E12830) | |
| **Subject Name** | Design and Analysis of Algorithm Lab |
| **Subject Code** | 20CSP-312 |
| **Branch** | Computer Science and Engineering |
| **Semester** | 5th |

**Experiment - 7**

**Student Name: Vivek Kumar UID: 21BCS8129**

**Branch: BE-CSE(LEET) Section/Group: 20BCS-WM-616/A**

**Semester: 5th Date of Performance: 31/10/2022**

**Subject Name: DAA Lab Subject Code: 20CSP-312**

1. **Aim/Overview of the practical:**

## Code to implement 0-1 Knapsack using Dynamic Programming

**2. Task to be done/ Which logistics used:**

Write a program to implement 0-1 Knapsack usingthe dynamic programming.

**3. Requirements (For programming-based labs):**

* Laptop or PC.
* Operation system (Mac, Windows, Linux, or any)
* Vs-Code with MinGw or any C++ Compiler

**4. Steps for experiment/practical/Code:**

#include <iostream>

#include <iostream>

using namespace std;

int max(int x, int y)

{

    return (x > y) ? x : y;

}

int knapSack(int W, int w[], int v[], int n)

{

    int i, wt;

    int K[n + 1][W + 1];

    for (i = 0; i <= n; i++)

    {

        for (wt = 0; wt <= W; wt++)

        {

            if (i == 0 || wt == 0)

                K[i][wt] = 0;

            else if (w[i - 1] <= wt)

                K[i][wt] = max(v[i - 1] + K[i - 1][wt - w[i - 1]], K[i - 1][wt]);

            else

                K[i][wt] = K[i - 1][wt];

        }

    }

    return K[n][W];

}

int main()

{

    cout << "Enter the number of items in a Knapsack: ";

    int n, W;

    cin >> n;

    int v[n], w[n];

    for (int i = 0; i < n; i++)

    {

        cout << "Enter value for item " << i << " : ";

        cin >> v[i];

        cout << "Enter weight for item " << i << " : ";

        cin >> w[i];

    }

    cout << "Enter the capacity of knapsack : ";

    cin >> W;

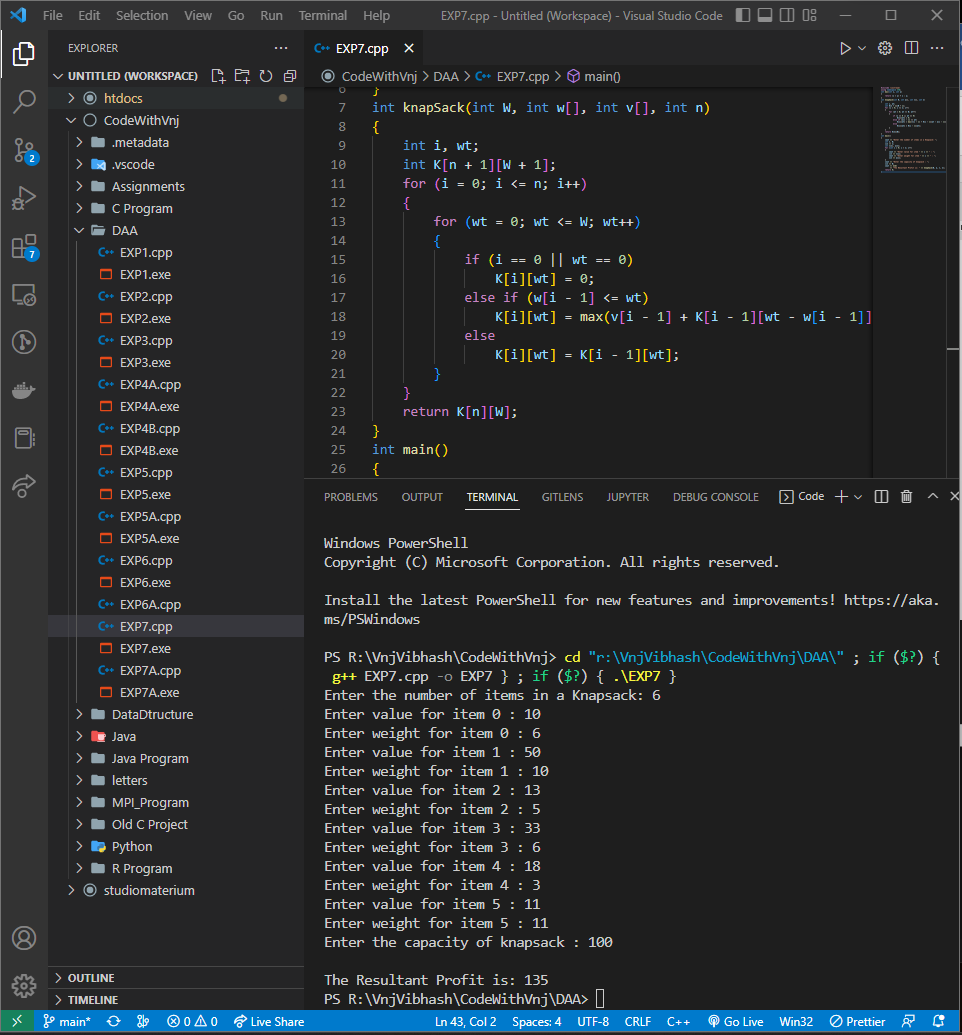
    cout << endl

         << "The Resultant Profit is: " << knapSack(W, w, v, n);

    return 0;

}

**5. Output:**

****

**Learning outcomes (What I have learnt):**

1. How to solve the 0-1 Knapsack using dynamic programming.

**Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Worksheet completion including writing learning objectives/Outcomes.  (To be submitted at the end of the day). |  |  |
| 2. | Post-Lab Quiz Result. |  |  |
| 3. | Student Engagement in  Simulation/Demonstration/Performance and Controls/Pre-Lab Questions. |  |  |
|  | Signature of Faculty (with Date): | Total Marks Obtained: |  |