**Practical-11**

**AIM:** To implement 0/1 Knapsack using Dynamic Programming in C language.

**SOFTWARE REQUIRED:** Vs Code

**PSEUDO CODE:**

function KnapsackDP(values[], weights[], n, W):

// Create a 2D array to store results

dp[n+1][W+1]

// Initialize the table with zeros

for i from 0 to n:

for w from 0 to W:

if i is 0 or w is 0:

dp[i][w] = 0

else if weights[i-1] <= w:

dp[i][w] = max(values[i-1] + dp[i-1][w-weights[i-1]], dp[i-1][w])

else:

dp[i][w] = dp[i-1][w]

// Create an array to track selected items

selected[n]

for i from n down to 1:

if i is 1 and dp[1][W] > 0:

selected[i] = 1

else if dp[i][W] != dp[i-1][W]:

selected[i] = 1

W = W - weights[i-1]

else:

selected[i] = 0

return dp[n][W], selected

**CODE:**

#include <stdio.h>

// Function to find the maximum of two integers

int max(int a, int b) {

    return (a > b) ? a : b;

}

int knapsack(int values[], int weights[], int n, int W, int selected[]) {

    int dp[n + 1][W + 1]; // Create a 2D array to store results

    // Initialize the table with zeros

    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 (weights[i - 1] <= w) {

                dp[i][w] = max(values[i - 1] + dp[i - 1][w - weights[i - 1]], dp[i - 1][w]);

                if (values[i - 1] + dp[i - 1][w - weights[i - 1]] > dp[i - 1][w]) {

                    selected[i] = 1; // Mark item as selected

                }

            } else {

                dp[i][w] = dp[i - 1][w];

            }

        }

    }

    return dp[n][W];

}

int main() {

    int n, W;

    printf("Ananta Walli, A2305221322");

    printf("\nEnter the number of items: ");

    scanf("%d", &n);

    int values[n], weights[n];

    for (int i = 0; i < n; i++) {

        printf("Enter the value and weight for item %d: ", i + 1);

        scanf("%d %d", &values[i], &weights[i]);

    }

    printf("Enter the maximum weight of the knapsack: ");

    scanf("%d", &W);

    int selected[n]; // Array to track selected items

    for (int i = 0; i < n; i++) {

        selected[i] = 0; // Initialize all items as not selected

    }

    int maxValue = knapsack(values, weights, n, W, selected);

    printf("Maximum value: %d\n", maxValue);

    printf("Selected items: ");

    for (int i = 0; i < n; i++) {

        if (selected[i]) {

            printf("Item %d, ", i + 1);

        }

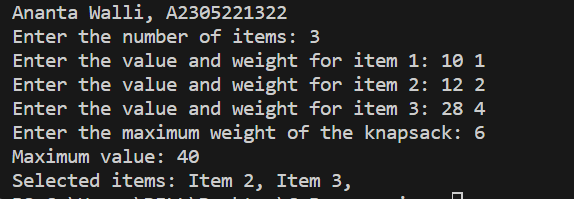
    }

    printf("\n");

    return 0;

}

**OUTPUT:**



**RESULT:** The above code implements 0/1 Knapsack using Dynamic Programming in C programming.