SANYAM AGRAWAL – SE21UCSE192 – CSE3

DAA Lab Assignment-8

01_Knapsack_DP.c

01_Knapsack_DP.c > knapsack(int, int [], int [], int)

```c
#include <stdio.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int knapsack(int max_capacity, int weights[], int profits[], int n) {
    int dp[n + 1][max_capacity + 1];

    // Build DP table
    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= max_capacity; w++) {
            if (i == 0 || w == 0) {
                dp[i][w] = 0;
            } else if (weights[i - 1] <= w) {
                dp[i][w] = max(profits[i - 1] + dp[i - 1][w - weights[i - 1]], dp[i - 1][w]);
            } else {
                dp[i][w] = dp[i - 1][w];
            }
        }
    }

    // Backtrack to find selected items
    int selected_items[n];
    int i = n, w = max_capacity;
    int num_selected = 0;
    while (i > 0 && w > 0) {
        if (dp[i][w] != dp[i - 1][w]) {
            selected_items[num_selected] = i - 1;
```

01_Knapsack_DP.c

```c
            w -= weights[i - 1];
            num_selected++;
        }
        i--;
    }

    printf("Selected Items: ");
    for (int j = num_selected - 1; j >= 0; j--) {
        printf("%d ", selected_items[j]);
    }
    printf("\n");

    return dp[n][max_capacity];
}

int main() {
    int profits[] = {2, 3, 1, 4};
    int weights[] = {3, 4, 6, 5};
    int max_capacity = 8;
    int n = sizeof(profits) / sizeof(profits[0]);

    int max_value = knapsack(max_capacity, weights, profits, n);
    printf("Maximum Value: %d\n", max_value);

    return 0;
}
```

Output Screenshot:->

```
sanyam@SANYAM:~/DAA$ gcc 01_Knapsack_DP.c -o out
sanyam@SANYAM:~/DAA$ ./out
Selected Items: 0 3
Maximum Value: 6
sanyam@SANYAM:~/DAA$
```