

0/1 Knapsack Problem using Dynamic Programming

```
#include <stdio.h>

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

int knapsack(int W, int wt[], int val[], int n) {
    int dp[n+1][W+1];

    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 (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 val[] = {60, 100, 120};
    int wt[] = {10, 20, 30};
    int W = 50;
    int n = sizeof(val) / sizeof(val[0]);
    printf("Maximum value in Knapsack = %d\n", knapsack(W, wt, val, n));
    return 0;
}
```

OUTPUT:

```
Maximum value in Knapsack = 220
Process returned 0 (0x0)   execution time : 0.009 s
Press any key to continue.
```