

## 0/1 KNAPSACK IMPLEMENTATION

```
#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];

    // Build dp table
    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= W; w++) {

            // Base case: no items or 0 capacity
            if (i == 0 || w == 0) {
                dp[i][w] = 0;
            }
            // If item can be included
            else if (wt[i-1] <= w) {
                dp[i][w] = max(val[i-1] + dp[i-1][w - wt[i-1]],
                                dp[i-1][w]);
            }
            // Otherwise, item cannot be included
            else {
                dp[i][w] = dp[i-1][w];
            }
        }
    }

    return dp[n][W]; // Final answer
}

int main() {
    int values[] = {60, 100, 120};
    int weights[] = {10, 20, 30};
    int capacity = 50;
    int n = sizeof(values) / sizeof(values[0]);
```

```
int result = knapsack(capacity, weights, values, n);
printf("Maximum value in Knapsack = %d\n", result);

return 0;
}
```

## OUTPUT

```
c:\Users\KIIT0001\DAALAB>cd "c:\Users\KIIT0001\DAALAB\" && gcc knapsack.c -o knapsack && "c:\Users\KIIT0001\DAALAB\knapsack"
Maximum value in Knapsack = 220
```