

# DAA ASSIGNMENT

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

            else
                dp[i][w] = max(dp[i-1][w],
                               val[i-1] + dp[i-1][w - wt[i-1]]);

        }
    }

    return dp[n][W];
}
```

```
int main() {
    int n, W;

    printf("Enter number of items: ");
    scanf("%d", &n);

    int wt[n], val[n];

    printf("Enter weights of items:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &wt[i]);

    printf("Enter values of items:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &val[i]);

    printf("Enter maximum capacity of knapsack: ");
    scanf("%d", &W);

    int result = knapsack(W, wt, val, n);

    printf("Maximum value in knapsack = %d\n", result);

    return 0;
}
```

```
C daaassignment.c ●
C daaassignment.c > knapsack(int, int [], int [], int)
1  #include <stdio.h>
2
3  int max(int a, int b) {
4      return (a > b) ? a : b;
5  }
6
7  int knapsack(int W, int wt[], int val[], int n) {
8      int dp[n+1][W+1];
9
10     for (int i = 0; i <= n; i++) {
11         for (int w = 0; w <= W; w++) {
12
13             if (i == 0 || w == 0)
14                 dp[i][w] = 0;
15
16             else if (wt[i-1] > w)
17                 dp[i][w] = dp[i-1][w];
18
19             else
20                 dp[i][w] = max(dp[i-1][w],
21                                val[i-1] + dp[i-1][w - wt[i-1]]);
22         }
23     }
24     return dp[n][W];
25 }
26
27 int main() {
28     int n, W;
29
30     printf("Enter number of items: ");
31     scanf("%d", &n);
32
33     int wt[n], val[n];
34 }
```

```

C daaassignment.c ●
C daaassignment.c > knapsack(int, int [], int [], int)
1   #include <stdio.h>
2
3   int max(int a, int b) {
4       return (a > b) ? a : b;
5   }
6
7   int knapsack(int W, int wt[], int val[], int n) {
8       int dp[n+1][W+1];
9
10      for (int i = 0; i <= n; i++) {
11          for (int w = 0; w <= W; w++) {
12
13              if (i == 0 || w == 0)
14                  dp[i][w] = 0;
15
16              else if (wt[i-1] > w)
17                  dp[i][w] = dp[i-1][w];
18
19              else
20                  dp[i][w] = max(dp[i-1][w],
21                                 val[i-1] + dp[i-1][w - wt[i-1]]);
22          }
23      }
24      return dp[n][W];
25  }
26
27  int main() {
28      int n, W;
29
30      printf("Enter number of items: ");
31      scanf("%d", &n);
32
33      int wt[n], val[n];
34

```

```

PS C:\Users\afifa\OneDrive\Desktop\random> cd "c:\Users\afifa\OneDrive\Desktop\random" ; if ($?) { gcc da
aassignment.c -o daaassignment } ; if ($?) { .\dhaarassignment }
c:/mingw/bin/../lib/gcc/mingw32/6.3.0/../../../libmingw32.a(main.o):(.,tex
nce to `WinMain@16'
collect2.exe: error: ld returned 1 exit status
PS C:\Users\afifa\OneDrive\Desktop\random> cd "c:\Users\afifa\OneDrive\De
aassignment.c -o daaassignment } ; if ($?) { .\dhaarassignment }
Enter number of items: 5
Enter weights of items:
50 60 70 80 90
Enter values of items:
100 200 225 222 111
Enter maximum capacity of knapsack: 150
Maximum value in knapsack = 447
PS C:\Users\afifa\OneDrive\Desktop\random>

```

 Snipping Tool

Screenshot copied to clipboard