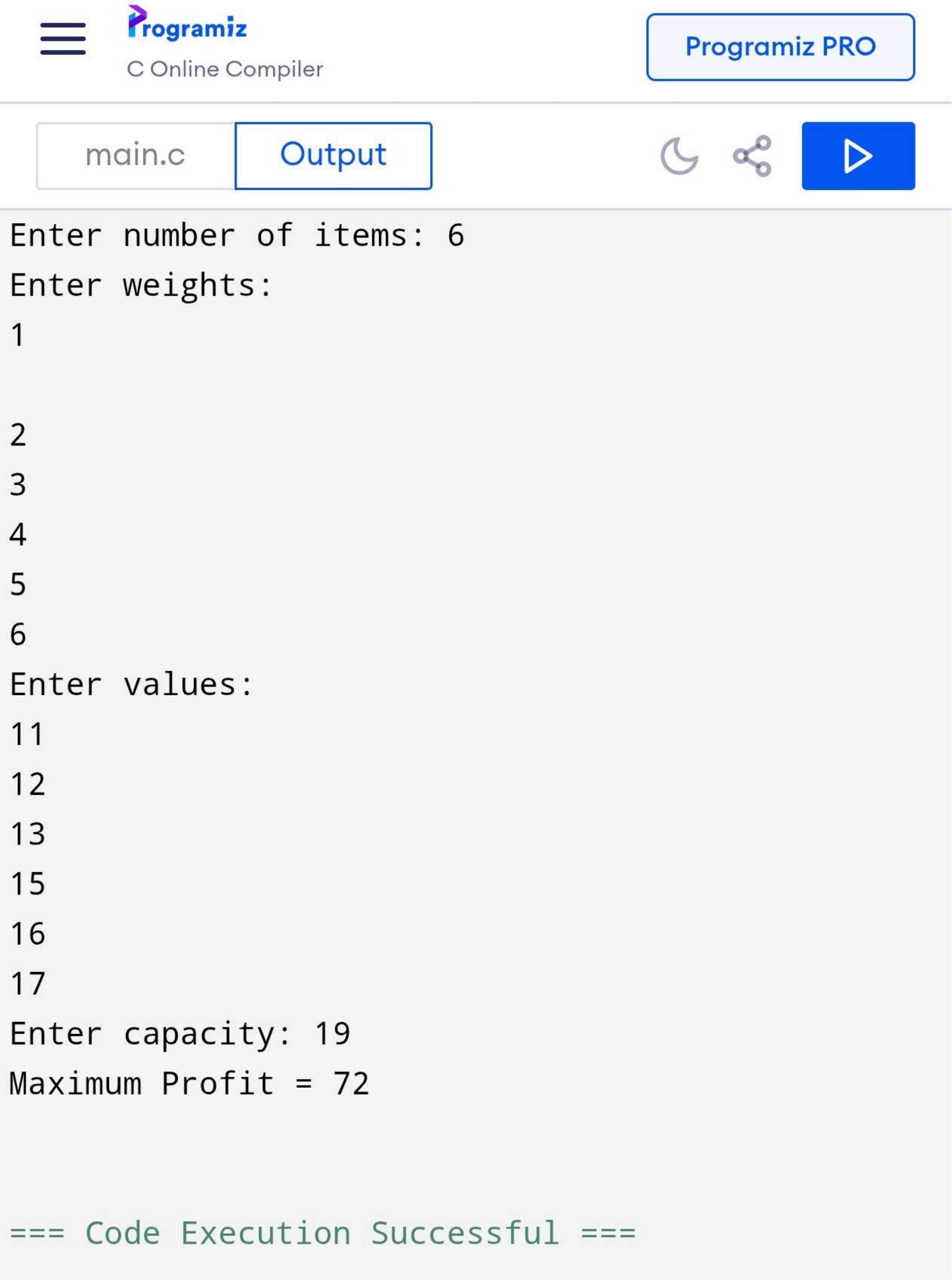
1. Dynamic Programming: Implement 0/1 Knapsack problem

```
#include <stdio.h>

int main(){
    int n, W, i, j;
    printf("Enter number of items: ");
    scanf("%d", &n);
    int w[n], v[n];
    printf("Enter weights:\n");
    for(i=0;i<n;i++) scanf("%d", &w[i]);
    printf("Enter values:\n");
    for(i=0;i<n;i++) scanf("%d", &v[i]);
    printf("Enter capacity: ");
    scanf("%d", &W);

    int dp[n+1][W+1];
    for(i=0;i<=n;i++){
        for(j=0;j<=W;j++){
            if(i==0||j==0) dp[i][j]=0;
            else if(w[i-1]<=j)
                dp[i][j] = (v[i-1] + dp[i-1][j-w[i-1]] > dp[i-1][j]) ?
                            v[i-1] + dp[i-1][j-w[i-1]] : dp[i-1][j];
            else dp[i][j] = dp[i-1][j];
        }
    }

    printf("Maximum Profit = %d\n", dp[n][W]);
    return 0;
}
```



```
Enter number of items: 6
Enter weights:
1

2
3
4
5
6
Enter values:
11
12
13
15
16
17
Enter capacity: 19
Maximum Profit = 72


=== Code Execution Successful ===
```