```java
import java.util.Scanner;

public class Knapsack {
    public void solve(int[] wt, int[] val, int W, int N) {
        int NEGATIVE_INFINITY = Integer.MIN_VALUE;
        int[][] m = new int[N + 1][W + 1];
        int[][] sol = new int[N + 1][W + 1];

        for (int i = 1; i <= N; i++) {
            for (int j = 0; j <= W; j++) {
                int m1 = m[i - 1][j];
                int m2 = NEGATIVE_INFINITY;
                if (j >= wt[i])
                    m2 = m[i - 1][j - wt[i]] + val[i];
                /** select max of m1, m2 **/
                m[i][j] = Math.max(m1, m2);
                sol[i][j] = m2 > m1 ? 1 : 0;
            }
        }

        /** make list of what all items to finally select **/
        int[] selected = new int[N + 1];
        for (int n = N, w = W; n > 0; n--) {
            if (sol[n][w] != 0) {
                selected[n] = 1;
                w = w - wt[n];
            }
            else
                selected[n] = 0;
        }

        /** Print finally selected items **/
        System.out.println("\nItems selected : ");
        for (int i = 1; i < N + 1; i++)
            if (selected[i] == 1)
                System.out.print(i +" ");
        System.out.println();
    }

    public static void main (String[] args) {
        Scanner scan = new Scanner(System.in);
```

```
42          System.out.println("Knapsack Algorithm Test\n");
43          Knapsack ks = new Knapsack();
44          System.out.println("Enter number of elements ");
45          int n = scan.nextInt();
46          int[] wt = new int[n + 1];
47          int[] val = new int[n + 1];
48
49          System.out.println("\nEnter weight for "+ n +" elements");
50          for (int i = 1; i <= n; i++)
51              wt[i] = scan.nextInt();
52          System.out.println("\nEnter value for "+ n +" elements");
53          for (int i = 1; i <= n; i++)
54              val[i] = scan.nextInt();
55
56          System.out.println("\nEnter knapsack weight ");
57          int W = scan.nextInt();
58          ks.solve(wt, val, W, n);
59      }
60  }
```