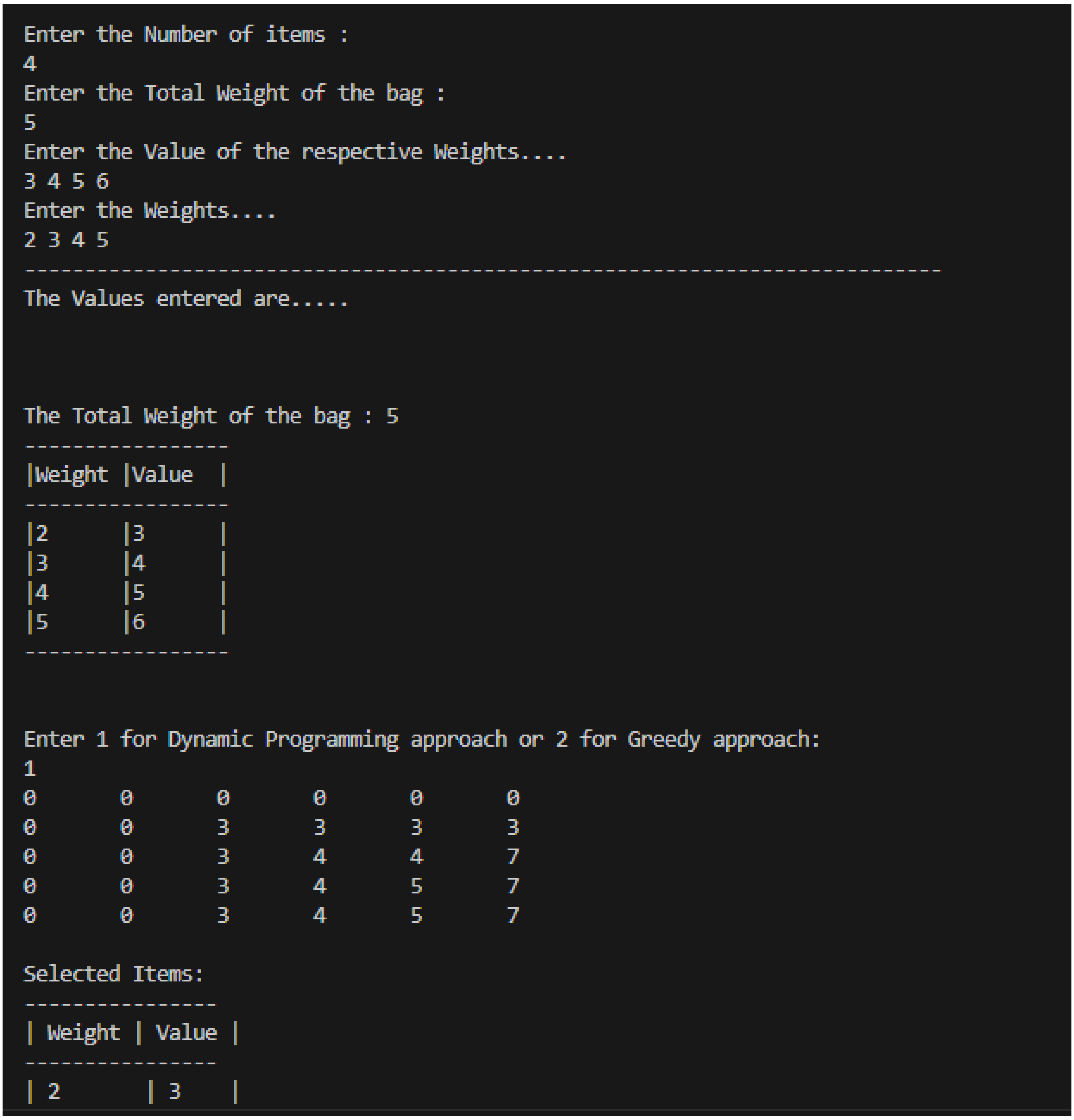
0/1 Knapsack

```
import java.util.*;


public class KnapSack01 {
    private static void knapSack(int weight, int[] weight2, int[] val, int len) {
        int K[][] = new int[len+1][weight+1];
        for (int i = 0; i <= len; i++) {
            for (int j = 0; j <= weight; j++) {
                if (i == 0 || j == 0) {
                    K[i][j] = 0;
                    System.out.print(K[i][j] + "\t"); // Use tabs for spacing
                } else if (weight2[i - 1] <= j) {
                    K[i][j] = Math.max(val[i - 1] + K[i - 1][j - weight2[i - 1]], K[i - 1][j]);
                    System.out.print(K[i][j] + "\t");
                } else {
                    K[i][j] = K[i - 1][j];
                    System.out.print(K[i][j] + "\t");
                }
            }
            System.out.println();
        }



        boolean[] selectedItems = new boolean[len];
        int i = len, j = weight;
        while (i > 0 && j > 0) {
            if (K[i][j] != K[i - 1][j]) {
                selectedItems[i - 1] = true;
                j -= weight2[i - 1];
            }
            i--;
        }
    // Print selected items
    System.out.println("\nSelected Items:");
    System.out.println("----------------");
    System.out.println("| Weight | Value |");
    System.out.println("----------------");
    for (i = 0; i < len; i++) {
        if (selectedItems[i]) {
            System.out.println("| " + weight2[i] + "\t  | " + val[i] + "\t |");
        }
    }
    System.out.println("----------------");
    System.out.println();
    System.out.println();
    System.out.println("The maximum Profit Generated : "+K[len][weight]);
}

private static void knapSackGreedy(int weight, int[] weight2, int[] val, int len) {
    // Greedy approach implementation
    double[] valuePerWeight = new double[len];
    for (int i = 0; i < len; i++) {
        valuePerWeight[i] = (double) val[i] / weight2[i];
    }

    double maxValue = 0;
    int[] selectedItems = new int[len];
    for (int i = 0; i < len; i++) {
        selectedItems[i] = 0;
    }

    while (weight > 0) {
        int maxIndex = -1;
```

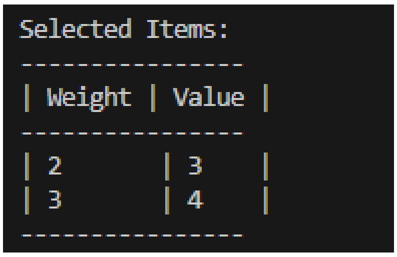
```
                double maxRatio = -1;
                for (int i = 0; i < len; i++) {
                    if (selectedItems[i] == 0 && valuePerWeight[i] > maxRatio) {
                        maxRatio = valuePerWeight[i];
                        maxIndex = i;
                    }
                }

                if (maxIndex == -1) {
                    break;
                }

                selectedItems[maxIndex] = 1;
                weight -= weight2[maxIndex];
                maxValue += val[maxIndex];
            }

            // Print selected items
            System.out.println("\nSelected Items (Greedy Approach):");
            System.out.println("----------------");
            System.out.println("| Weight | Value |");
            System.out.println("----------------");
            for (int i = 0; i < len; i++) {
                if (selectedItems[i] == 1) {
                    System.out.println("| " + weight2[i] + "\t  | " + val[i] + "\t |");
                }
            }
            System.out.println("----------------");
            System.out.println("The maximum Profit Generated (Greedy): " + maxValue);
        }
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
```

```java
System.out.println("Enter the Number of items : ");

int n = sc.nextInt();

int val[] = new int[n];

int weight[] = new int[n];

System.out.println("Enter the Total Weight of the bag : ");

int Weight = sc.nextInt();


System.out.println("Enter the Value of the respective Weights....");

for(int i = 0;i<val.length;i++){

    val[i] = sc.nextInt();

 }


System.out.println("Enter the Weights....");

for(int i = 0;i<weight.length;i++){

     weight[i] = sc.nextInt();

}


System.out.println("------------------------------------------------------------------------------------------");

System.out.println("The Values entered are.....\n");

System.out.println();

System.out.println();

System.out.println("The Total Weight of the bag : "+Weight);

System.out.println("-----------------");

System.out.println("|Weight\t|Value\t|");

System.out.println("-----------------");

for(int  i =0;i<weight.length;i++){

   System.out.println("|"+weight[i]+"\t|"+val[i]+"\t|");

}

System.out.println("-----------------");

System.out.println();

System.out.println();

int len = val.length;
```

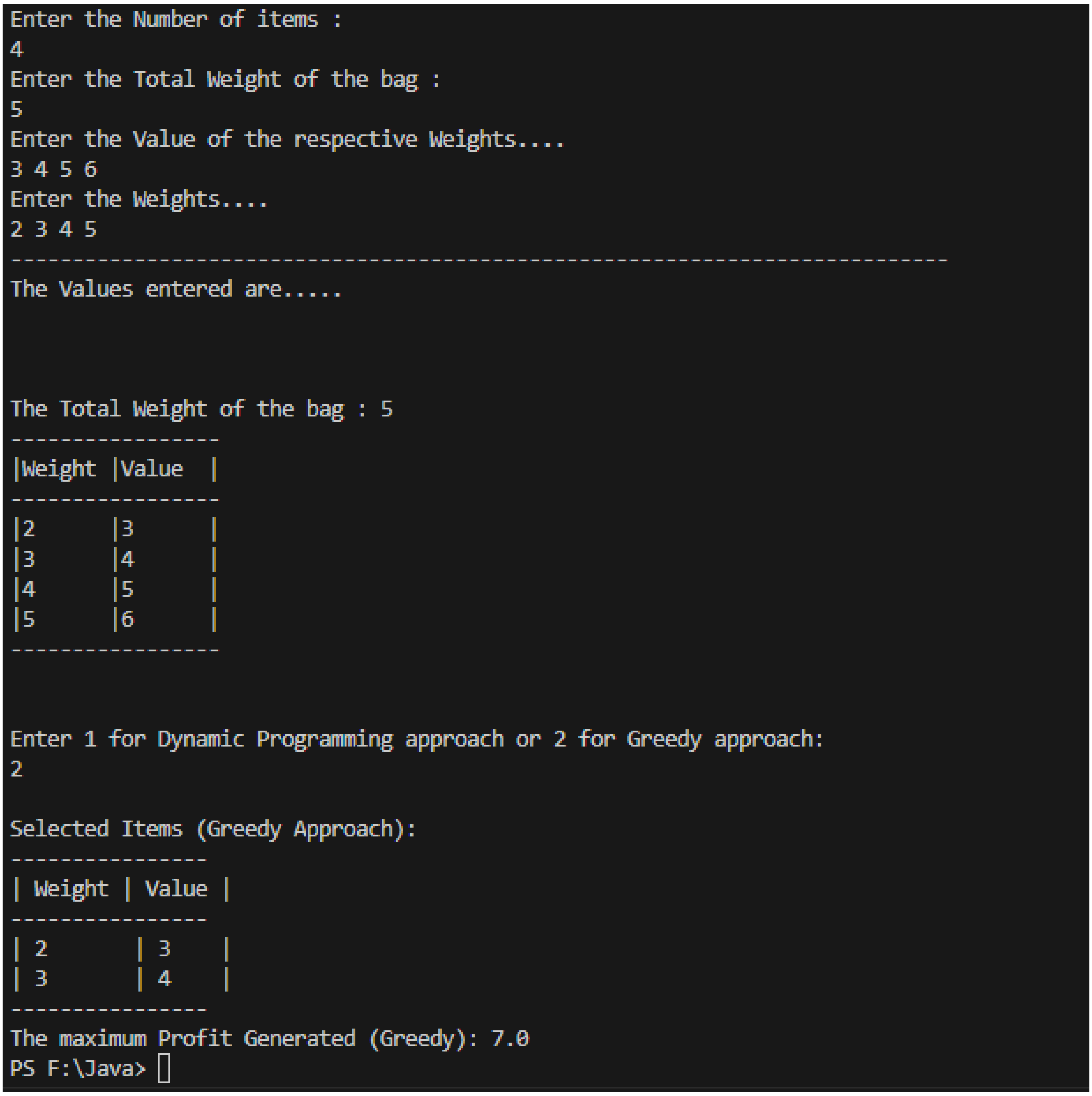
```
        int choice;
        System.out.println("Enter 1 for Dynamic Programming approach or 2 for Greedy approach:");
        choice = sc.nextInt();


        switch (choice) {
            case 1:
                knapSack(Weight, weight, val, len);
                break;
            case 2:
                knapSackGreedy(Weight, weight, val, len);
                break;
            default:
                System.out.println("Invalid choice");
        }

    }


}
```



```
Enter the Number of items :
4
Enter the Total Weight of the bag :
5
Enter the Value of the respective Weights....
3 4 5 6
Enter the Weights....
2 3 4 5
----------------------------------------------------------------------------
The Values entered are.....



The Total Weight of the bag : 5
-----------------
|Weight |Value  |
-----------------
|2      |3      |
|3      |4      |
|4      |5      |
|5      |6      |
-----------------


Enter 1 for Dynamic Programming approach or 2 for Greedy approach:
1
0       0       0       0       0       0
0       0       3       3       3       3
0       0       3       4       4       7
0       0       3       4       5       7
0       0       3       4       5       7

Selected Items:
----------------
| Weight | Value |
----------------
| 2        | 3    |
```



```
Selected Items:
----------------
| Weight | Value |
----------------
| 2        | 3    |
| 3        | 4    |
----------------
```



```
Enter the Number of items :
4
Enter the Total Weight of the bag :
5
Enter the Value of the respective Weights....
3 4 5 6
Enter the Weights....
2 3 4 5
----------------------------------------------------------------------------
The Values entered are.....



The Total Weight of the bag : 5
-----------------
|Weight |Value  |
-----------------
|2      |3      |
|3      |4      |
|4      |5      |
|5      |6      |
-----------------


Enter 1 for Dynamic Programming approach or 2 for Greedy approach:
2

Selected Items (Greedy Approach):
----------------
| Weight | Value |
----------------
| 2       | 3    |
| 3       | 4    |
----------------
The maximum Profit Generated (Greedy): 7.0
PS F:\Java> 
```