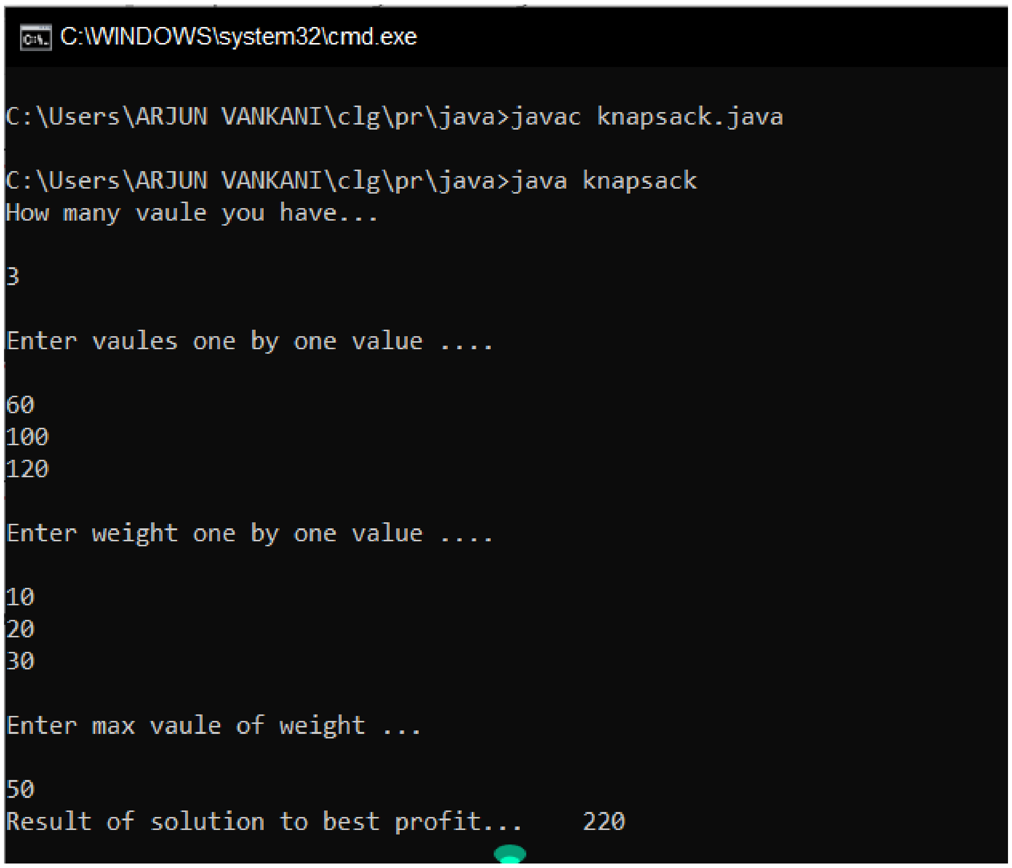
Practical 10: WAP for Knapsack Problem using Dynamic programming.

Program:

```
import java.util.Scanner;

public class knapsack {
            static int max(int a, int b)
   {
      return (a > b) ? a : b;
   }

            static int knapSack(int W_max, int weight[],int val[], int num)
   {

      if (num == 0 || W_max == 0)
         return 0;

      if (weight[num - 1] > W_max)

         return knapSack(W_max, weight, val, num-1);
      else
         return max(val[num - 1]+knapSack(W_max - weight[num - 1], weight, val, num-1),
            knapSack(W_max, weight, val, num-1));
```

```
    }

        public static void main(String[] args) {



            Scanner n = new Scanner(System.in);

            System.out.println("How many vaule you have...\n");
            int num = n.nextInt();
            int val[]=new int[num];
            int weight[]=new int[num];
            System.out.println("\nEnter vaules one by one value ....\n");
            for(int i=0;i<num;i++)
                {
                    val[i]= n.nextInt();
                }
            System.out.println("\nEnter weight one by one value ....\n");
            for(int i=0;i<num;i++)
                {
                    weight[i]= n.nextInt();
                }
            System.out.println("\nEnter max vaule of weight ...\n ");
            int W_max = n.nextInt();
      System.out.println("Result of solution to best profit...\t"+knapSack(W_max, weight, val, num));
        }

}
```

Output:



```
C:\WINDOWS\system32\cmd.exe

C:\Users\ARJUN VANKANI\clg\pr\java>javac knapsack.java

C:\Users\ARJUN VANKANI\clg\pr\java>java knapsack
How many vaule you have...

3

Enter vaules one by one value ....

60
100
120

Enter weight one by one value ....

10
20
30

Enter max vaule of weight ...

50
Result of solution to best profit...    220
```