KARIYA RAJ 21162101011

ALGORITHM ANALYSIS AND DESIGN

PRACTICAL -11

You are working at the cash counter at a fun-fair, and you have three types: A thief carrying a single knapsack with limited (W = 5) capacity. The museum you stole had (n=4) artifacts that you could steal. Unfortunately, you might not be able to steal the entire artifact because of your limited knapsack capacity.

Help the thief to cherry pick the artifact in order to maximize the total value =W)

of the artifacts you stole.

First solve the given below example:

Let n = 4, W=5

(P1, P2, P3, P4) = (3,4,5,6)

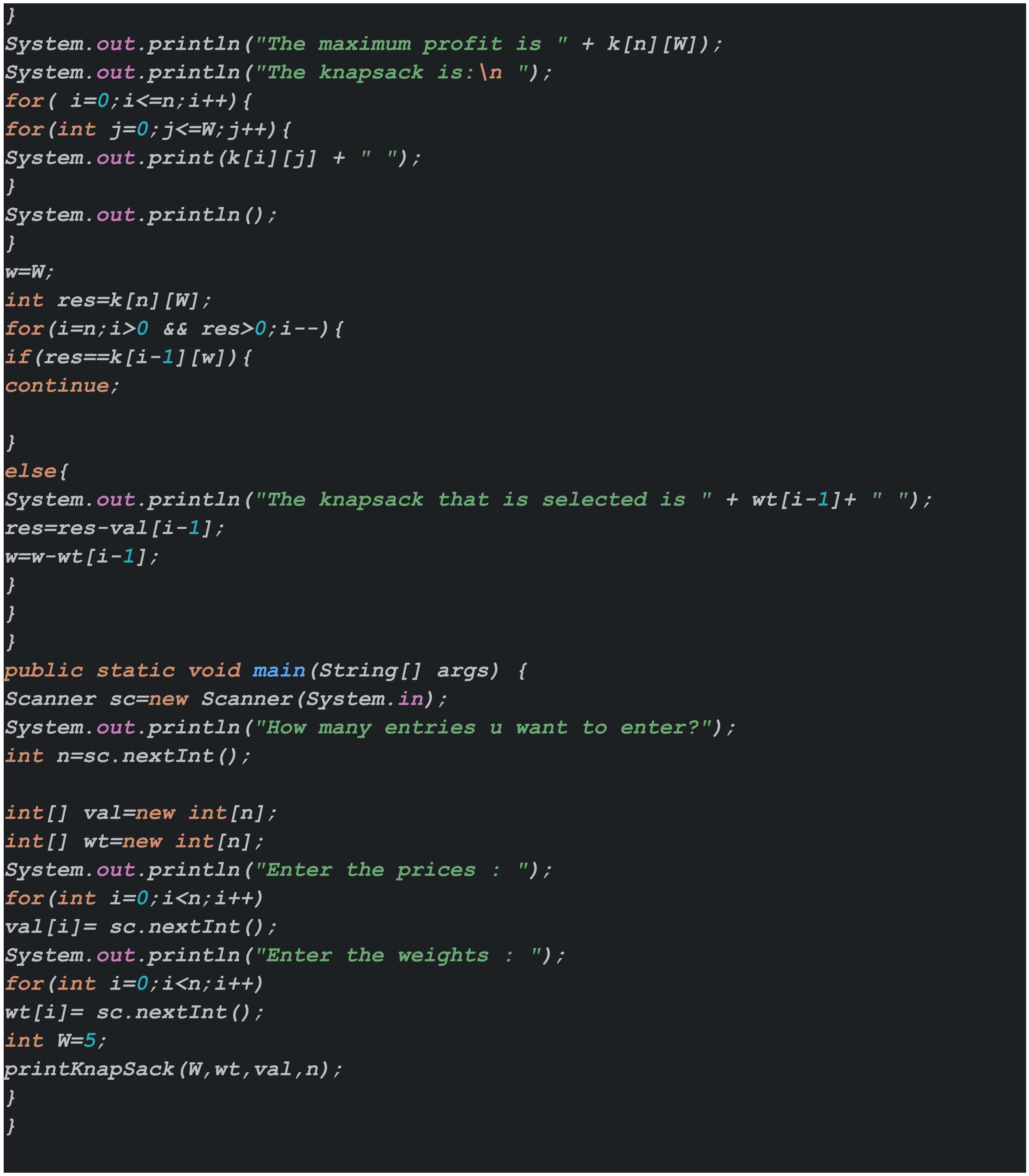
(w1, w2, w3, w4) = (2,3,4,5)

CODE:

```
package aad;
import java.lang.*;
import java.util.Scanner;

public class knapsack {
static void printKnapSack(int W,int[] wt,int[] val,int n){
int i,w;
int[][] k=new int[n+1][W+1];

for(i=0;i<=n;i++){
for(w=0;w<=W;w++){
if(i==0 || w==0)
k[i][w]=0;
else if(wt[i-1]<=w){
k[i][w]=Math.max(val[i-1]+k[i-1][w-wt[i-1]],k[i-1][w]);
}else{
k[i][w]=k[i-1][w];
}
}
```



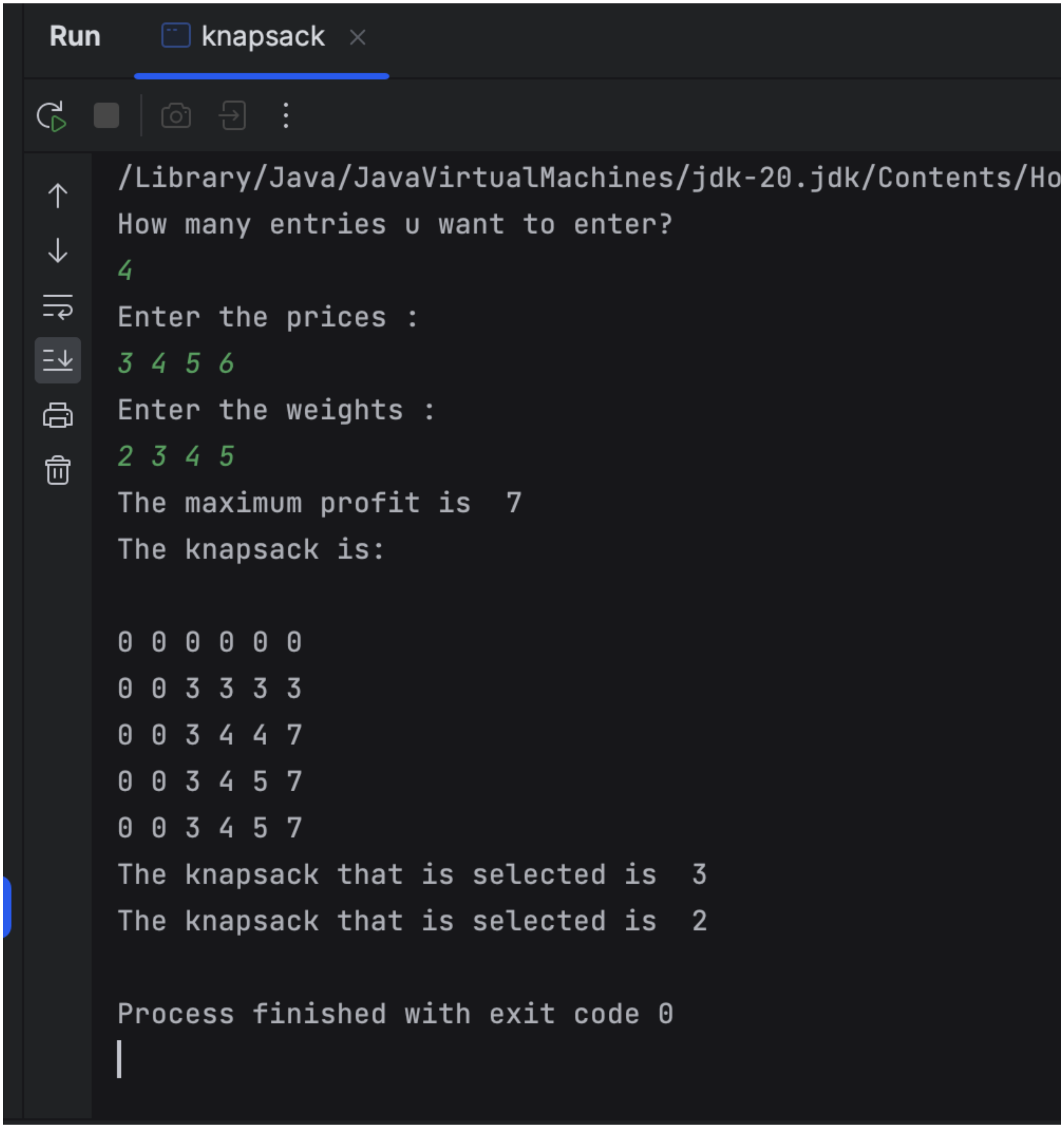
```
}
System.out.println("The maximum profit is " + k[n][W]);
System.out.println("The knapsack is:\n ");
for( i=0;i<=n;i++){
for(int j=0;j<=W;j++){
System.out.print(k[i][j] + " ");
}
System.out.println();
}
w=W;
int res=k[n][W];
for(i=n;i>0 && res>0;i--){
if(res==k[i-1][w]){
continue;

}
else{
System.out.println("The knapsack that is selected is " + wt[i-1]+ " ");
res=res-val[i-1];
w=w-wt[i-1];
}
}
}
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("How many entries u want to enter?");
int n=sc.nextInt();

int[] val=new int[n];
int[] wt=new int[n];
System.out.println("Enter the prices : ");
for(int i=0;i<n;i++)
val[i]= sc.nextInt();
System.out.println("Enter the weights : ");
for(int i=0;i<n;i++)
wt[i]= sc.nextInt();
int W=5;
printKnapSack(W,wt,val,n);
}
}
```

OUTPUT:

KARIYA RAJ 21162101011



```
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Ho
How many entries u want to enter?
4
Enter the prices :
3 4 5 6
Enter the weights :
2 3 4 5
The maximum profit is  7
The knapsack is:


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
The knapsack that is selected is  3
The knapsack that is selected is  2

Process finished with exit code 0
```