```
      BufferedReader br;
       File archivo = new File("entrada.txt");
       if (archivo.exists()) br = new BufferedReader(new FileReader(archivo));
       else br = new BufferedReader(new InputStreamReader(System.in));
```

Exact Change

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String[] args) throws Exception {
LECTURA TXT.
        StringBuilder sb = new StringBuilder();
        StringTokenizer st;
        int casos = Integer.parseInt(br.readLine());
        while(casos-- > 0){
             int R = Integer.parseInt(br.readLine()), n = Integer.parseInt(br.readLine());
             int[] coins = new int[n];
             for (int i = 0; i < n; i++) {
                         coins[i] = Integer.parseInt(br.readLine());
                   }
             long[]A = new long[10001];
             Arrays.fill(A, Integer.MAX_VALUE);
             A[0] = 0;
             for (int i = 0; i < n; i++) {
                   int taken = coins[i];
                         for (int v = 10000; v >=0; v--) {
                               if(A[v] != Integer.MAX_VALUE && v+taken<=10000){
                                     A[v+taken] = Math.min(A[v+taken], A[v]+1);
                               }
                         }
                   }

             for (int j = R; j < 10001; j++) {
                   if (A[j] != Integer.MAX_VALUE) {
                         sb.append(j).append(" ").append(A[j]).append("\n");
                   break;
                   }
             }
        }
        System.out.print(sb);
    }
}
```

The jackpot

```
import java.io.*;
import java.util.*;
public class Main {
       public static int[] vec;
       public static int[] dp;
    public static void main(String[] args) throws Exception {
       Scanner in = new Scanner(System.in);
        int n;
        StringBuilder sb = new StringBuilder();
        while((n = in.nextInt()) != 0){
             vec = new int[10000 + 10];
             dp = new int[10000 + 10];
             for (int i = 0; i < n; i++) {
                         vec[i] = in.nextInt();
                   }
             long best = -1;
             for (int k = 1; k < n; k++) {
```

```
                dp[k] =vec[k];
                if(dp[k-1]+vec[k] > vec[k]) dp[k] =dp[k-1]+vec[k];
            }
            for (int k = 1; k < n; k++) {
                if(dp[k]>best) best = dp[k];
            }
            if(best>0) sb.append("The maximum winning streak is ").append(best).append(".\n");
            else sb.append("Losing streak.\n");
        }
        System.out.print(sb);
    }
}
```

Dividing coins

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String[] args) throws Exception {
        LECTURA TXT
        StringBuilder sb = new StringBuilder();
        StringTokenizer st;
        int casos = Integer.parseInt(br.readLine());
        while(casos-- > 0){
            int n = Integer.parseInt(br.readLine()), sum =0;
            int[] coins = new int[n+1];
            st= new StringTokenizer(br.readLine());
            for (int i = 1; i <= n; i++) {
                    coins[i] = Integer.parseInt(st.nextToken());
                    sum+= coins[i];
                }
            int[][] A = new int[n+1][sum+1];
            for (int i = 1; i <= n; i++) {
                    for (int k = 1; k <= sum; k++) {
                        int op1 = A[i-1][k], op2 = Integer.MIN_VALUE;
                        if (coins[i] <= k) op2 = coins[i] + A[i-1][k-coins[i]];
                        A[i][k] = Math.max(op1,op2);
                    }
                }
            int res = Integer.MAX_VALUE;
           for (int i = 0; i <= sum; ++i)
               res = Math.min(res, Math.abs((sum - A[n][i]) - A[n][i]));
           System.out.println(res);
        }
    }
}
```

SuperSale

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String[] args) throws Exception {
        Lectura TXT

        StringBuilder sb = new StringBuilder();
        StringTokenizer st;
        int casos = Integer.parseInt(br.readLine());
        while(casos-- > 0){
            int n = Integer.parseInt(br.readLine());
            int[] precio = new int[n+1], peso = new int[n+1];
            for (int i = 1; i <= n; i++) {
                st= new StringTokenizer(br.readLine());
                    precio[i] = Integer.parseInt(st.nextToken());
                    peso[i] = Integer.parseInt(st.nextToken());
                }
```

```
            int sum = 0;
            int m = Integer.parseInt(br.readLine());
            for(int p=0; p<m; p++){
                  int R = Integer.parseInt(br.readLine());
                  int[][] A = new int[n+1][R+1];
                  for (int i = 1; i <= n; i++) {
                        for (int k = 1; k <= R; k++) {
                              int op1 = A[i-1][k], op2 = Integer.MIN_VALUE;
                              if (peso[i] <= k) op2 = precio[i] + A[i-1][k-peso[i]];
                              A[i][k] = Math.max(op1,op2);
                        }
                  }
                  sum += A[n][R];
            }
            sb.append(sum).append("\n");
      }
      System.out.print(sb);
   }
}
```

Let Me Count The Ways

```
import java.io.*;
import java.util.*;
public class Main {
      public static int[] coins = {50, 25, 10, 5, 1};
      public static long[] ways = new long[30000+5];
   public static void main(String[] args) throws Exception {
      LECTURA TXT

       String ent;
       StringBuilder sb = new StringBuilder();
       StringTokenizer st;
       solve();
       while((ent=br.readLine()) != null){
            int m = Integer.parseInt(ent);
            long n = ways[m];
            if(n != 1) sb.append("There are ").append(n).append(" ways to produce
").append(m).append(" cents change.\n");
            else sb.append("There is only 1 way to produce ").append(m).append(" cents
change.\n");
       }
       System.out.print(sb);
   }
   public static void solve(){
      ways[0] = 1;
      int taken;
      for (int i = 0; i < 5; i++) {
            taken = coins[i];
                  for (int k = taken; k < 30005; k++) {
                        ways[k] += ways[k-taken];
                  }
            }
   }
}
```

Shoemaker's Problem

```
import java.io.*;
import java.util.*;
public class Main {
    public static <T> void main(String[] args) throws Exception {
        LECTURA TXT
```

```
        int casos = Integer.parseInt(br.readLine());
        StringBuilder sb = new StringBuilder();
        StringTokenizer st;
        br.readLine();
        for (int c = 0; c < casos; c++) {
             int n = Integer.parseInt(br.readLine());
             double[][] vec = new double[n][n];
             for (int i = 0; i < n; i++) {
                    st = new StringTokenizer(br.readLine());
                    int t = Integer.parseInt(st.nextToken()), s = Integer.parseInt(st.nextToken());
                           vec[i] = new double[]{1.*s/t, i+1};
                    }
             Arrays.sort(vec, new Comparator<double[]>() {
                           public int compare(double[] o1, double[] o2) {
                                  if(o1[0] != o2[0]) return (o2[0] < o1[0]) ? -1 : 1;
                                  return (""+o1[1]).compareTo(""+o2[1]);
                           }
                    });
             for (int i = 0; i < n; i++) {
                           if(i!=0) sb.append(" ");
                    sb.append((int)vec[i][1]);
                    }
             br.readLine();
             sb.append("\n");
             if(c<casos-1) sb.append("\n");
        }
        System.out.print(sb);
    }
}
```