Chrystal Mingo

CSC 22000

Professor Lucci

Methodology of Project 2

The **postage stamp problem** is a mathematical riddle that asks what is the smallest postage value which cannot be placed on an envelope, if the letter can hold only a limited number of stamps, and these may only have certain specified face values.

```
import java.util.ArrayList;
import java.util.Scanner;
public class stampProblem {

  public static void main(String[] args) {

    Scanner keyboard = new Scanner(System.in);
    ArrayList<Integer> denoms = new ArrayList<Integer>();

    // Get user input
    System.out.println("How many stamps fit on an envelope? ");
    int n = keyboard.nextInt();
    String junk = keyboard.nextLine(); //Have to flush the end-of-line character
    // Get Denomination Info.
    System.out.println("What denominations are available? (in ascending order, all on one line): ");
    Scanner line = new Scanner(keyboard.nextLine());

    int biggest=1;
    // This loop adds N denominations given to an arraylist called denoms
    while(line.hasNextInt()){
      biggest = line.nextInt();
      denoms.add(biggest);
    }
    // This loop will check for all the amounts which can be made starting from amount = 0 to amount = biggestdnomination * n
    for(int i=0; i<=n*biggest; i++){
      //
      ArrayList<Integer> soFar = new ArrayList<>();

      // Print output.
```

```
        System.out.println("Make "+i+" from "+n+" of "+denoms+": "+canMake(i,n,denoms)+"
"+canMake(i,n,denoms,soFar));
      }
   }

   // Can Make is a recursive method that will solve the postage stamp problem.
   // Explanation for recusrsion :
   // Base cases is when amount is 0 then number of stamps should be greater than or equal to 0. This will denote a valid case.
   // Else we try to subtract every possible denominations from current amount and also decrease the stamps by 1  and check
using recusrion if it is possible to make it.
   // If amount recieved is negative or stamps becomes negative then we cant make that amount in that in that case hence we
return false.
   public static boolean canMake(int amount, int stamps, ArrayList<Integer> denominations){
      // Complete this here:
      if(amount == 0 && stamps >= 0)
         return true;
      if(amount < 0 || stamps < 0)
         return false;
      for(int val : denominations){
         if(canMake(amount - val, stamps-1, denominations)){
            return true;
         }
      }
      return false;
   }

   // The returned Arralist will have the list of denomination which will tell denom used to create the amount
   public static ArrayList<Integer> canMake(int amount, int stamps, ArrayList<Integer> denominations,ArrayList<Integer>
soFar){
      //System.out.println("Checking "+amount+"/"+stamps+"/"+denominations);
      if(amount == 0 && stamps >= 0)
         return new ArrayList<Integer>();
      if(amount < 0 || stamps < 0)
         return null;
      for(int val : denominations){
         ArrayList<Integer> arr = canMake(amount - val, stamps-1, denominations,soFar);
         if(arr != null){
            arr.add(val);
            return arr;
```

```
            }
        }
        return null;
    }

}
```