**package** AlgorithmDpPractise;

**class** coinChangeDP

{

**public int**[] **coins**;

**public int sum**;

**public int result**;

**public int**[] **memorization**;

**public int**[] **check**;

*//create a constructor to store the coins and sum which is comes from main method*

**public** coinChangeDP(**int**[] coins, **int** sum) {

**this**.**coins** = coins;

**this**.**sum** = sum;

**this**.**memorization** = **new int**[sum+1];

**this**.**check** = **new int**[sum + 1];

}

**public void** initialize()

{

**for** (**int** i = 0; i <=**sum** ; i++) {

**memorization**[i] = 0;

**check**[i] = 0;

}

}

**public int** findMinimumNumberCoins(**int** sum)

{

**if** (sum<0) **return** 9999;

**if** (sum == 0) **return** 0;

**if** (**check**[sum]!=0)

{

**return memorization**[sum];

}

**else**

{

**result** = 999999;

**for** (**int** i=0;i<**coins**.**length**;i++)

{

**result** = Math.*min*(**result**,findMinimumNumberCoins(sum-**coins**[i])+1);

}

**check**[sum] = 1;

**memorization**[sum] = **result**;

**return memorization**[sum];

}

}

}

**public class** minNumberOfWaysCoinUsingRecursion {

**public static void** main(String[] args) {

**int**[] coins = {1, 2};

**int** sum = 3;

*//create a object for call that class*

coinChangeDP object = **new** coinChangeDP(coins,sum);

object.initialize();

System.***out***.println(**"Minimum number of coins required = "**+object.findMinimumNumberCoins(sum));

}

}