**Problem Marks (8)**

PayMoney. processes thousands of transactions daily amounting to crores of Rupees. They also have a daily target that they must achieve. Given a list of transactions done by PayMoney and a daily target, your task is to determine at which transaction PayMoney achieves the same. If the target is not achievable, then display the target is not achieved.

**TestCase 1**

enter the size of transaction array

3

enter the values of array

20 12 31

enter the total no of targets that needs to be achieved

2

enter the value of target

21

Target achieved after 2 transactions

enter the value of target

19

Target achieved after 1 transactions

**Explanation**

Target 1 i.e 21 is achieved after 2 transactions, (20 + 12)

Target 2 i.e 19 is achieved in the 1st transaction itself.

-------------------------------------------------------

**Test Case 2**

enter the size of transaction array

1

enter the values of array

100

enter the total no of targets that needs to be achieved

1

enter the value of target

101

Given target is not achieved

**Explanation** → Since there is only 1 transaction that is of 100 and the target value is 101, hence target is not achieved.

**Solution →**

package com.greatlearning.iitr.dsa.labsession2;

import java.util.Scanner;

class Transaction {

public static void main(String args[]) throws Exception {

Scanner s = new Scanner(System.in);

System.out.println("enter the size of transaction array");

int size = s.nextInt();

int arr[] = new int[size];

System.out.println("enter the values of array");

for (int i = 0; i < size; i++)

arr[i] = s.nextInt();

System.out.println("enter the total no of targets that needs to be achieved");

int targetNo = s.nextInt();

while (targetNo-- != 0) {

int flag = 0;

long target;

System.out.println("enter the value of target");

target = s.nextInt();

long sum = 0;

for (int i = 0; i < size; i++) {

sum += arr[i];

if (sum >= target) {

System.out.println("Target achieved after "+(i + 1) + " transactions ");

flag = 1;

break;

}

}

if (flag == 0) {

System.out.println(" Given target is not achieved ");

}

}

}

}

**-----------------------------------------------------------------------------------------------------------**

**Question**   **Marks (12 )**

You are a traveler and traveling to a country where the currency denominations are unknown and as you travel you get to know about the denomination in random order.

You want to make a payment of amount x, in such a way that the higher denomination is used to make exact payment.

**Input**

Take input of all the currency denominations ( random order)

Take input of the amount that you want to pay.

**Output**

Print the minimum no of notes that you will be using to pay the net amount.

**TestCase 1**

enter the size of currency denominations

3

enter the currency denominations value

5

1

10

enter the amount you want to pay

12

Your payment approach in order to give min no of notes will be

10:1

1:2

**TestCase 2**

enter the size of currency denominations

5

enter the currency denominations value

60

5

12

78

25

enter the amount you want to pay

128

Your payment approach in order to give min no of notes will be

78:1

25:2

**TestCase 3**

enter the size of currency denominations

4

enter the currency denominations value

12

5

123

18

enter the amount you want to pay

158

Your payment approach in order to give min no of notes will be

123:1

18:1

12:1

5:1