**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**I MCA**

**20MX16 DATA STRUCTURES LABORATORY**

**Problem Sheet -1**

**Date: 29-09-2022**

1. Write a program to keep asking for a number less than 100 and stop when the number is multiple of 3, 5, and 7
2. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Write a program to find the sum of all the multiples of 3 or 5 below 1000.

Hint: Be careful not to add two times a number which is multiple of both 3 and 5.

1. Write while loops to do the following:

– Repeatedly print the value of the variable xValue, decreasing it by 0.5 each time, as long as xValue remains positive.

– Print the square roots of the first 25 odd positive integers.

1. Write a program to print the following

|  |  |  |
| --- | --- | --- |
| **1       222     33333   4444444 555555555** |  | **1       212     32123   4321234 543212345** |
| **Challenging Question (CodeChef)** |  |  |

1. There are **N** students living in the dormitory of Berland State University. Each of them sometimes wants to use the kitchen, so the head of the dormitory came up with a timetable for kitchen's usage in order to avoid the conflicts:

* The first student starts to use the kitchen at the time **0** and should finish the cooking not later than at the time **A1**.
* The second student starts to use the kitchen at the time **A1** and should finish the cooking not later than at the time **A2**.
* And so on.
* The **N**-th student starts to use the kitchen at the time **AN-1** and should finish the cooking not later than at the time **AN**

The holidays in Berland are approaching, so today each of these **N** students wants to cook some pancakes. The **i**-th student needs **Bi** units of time to cook.

The students have understood that probably not all of them will be able to cook everything they want. How many students will be able to cook without violating the schedule?

**Input**

The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows.

The first line of each test case contains a single integer **N** denoting the number of students.

The second line contains **N** space-separated integers **A1**, **A2**, ..., **AN** denoting the moments of time by when the corresponding student should finish cooking.

The third line contains **N** space-separated integers **B1**, **B2**, ..., **BN** denoting the time required for each of the students to cook.

**Output**

For each test case, output a single line containing the number of students that will be able to finish the cooking.

**Constraints**

Should contain all the constraints on the input data that you may have. Format it like:

* **1** ≤ **T** ≤ **10**
* **1** ≤ **N** ≤ **104**
* **0** < **A1** < **A2** < ... < **AN** < **109**
* **1** ≤ **Bi** ≤ **109**

**Example**

**Input:**

2

3

1 10 15

1 10 3

3

10 20 30

15 5 20

**Output:**

2

1

**Explanation**

**Example case 1.** The first student has **1** unit of time - the moment **0**. It will be enough for her to cook. The second student has **9** units of time, but wants to cook for **10** units of time, and won't fit in time. The third student has **5** units of time and will fit in time, because needs to cook only for **3** units of time.

**Example case 2.** Each of students has **10** units of time, but only the second one will be able to fit in time.