

**Problem** Test Cases OutputQ:
01/02

1

2

The current selected programming language is **Java**. We emphasize the submission of a **working code** over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use `System.out.println()` to debug your code. The `System.out.println()` may not work in case of syntax/runtime error. The version of **JDK** being used is **1.8**.

In Amazon's sort center, a computer system decides what packages are to be loaded on what truck. All rooms and spaces are abstracted into space units which is represented as an integer. For each type of truck, they have different space units. For each package, they will be occupying different space units.

As a software development engineer in sort centers, you will need to write a method. Given truck space units and a list of product space units, find out exactly TWO products that fit into the truck. You will also implement an internal rule that the truck has to reserve exactly 30 space units for safety purposes. Each package is assigned a unique ID, numbered from 0 to N-1.

Assumptions:

1. You will pick up exactly 2 packages.
2. You cannot pick up one package twice.
3. If you have multiple pairs, select the pair with the largest package.

Input

The input to the function/method consists of two arguments -
truckSpace, an integer representing the truck space.

packagesSpace, a list of integers representing the space units occupying by packages.

Output

Return a list of integers representing the IDs of two packages whose combined space will leave exactly 30 space units on the truck.

Example

Input:

truckSpace = 90

packagesSpace = [1, 10, 25, 35, 60]

Output:

[2, 3]

Explanation:

Given a truck of 90 space units, a list of packages space units [1, 10, 25, 35, 60], Your method should select the third(ID-2) and fourth(ID-3) package since you have to reserve exactly 30 space units.