# Problem 3 – Ticket Trouble

Our hero finally managed to park successfully and now realized that he has forgotten to clear out his suitcase and has a bunch of documents and old plane tickets in there. He doesn’t have enough time to go through it all and needs your help to find the tickets for the trip. Sam can tell you the location they’re headed and give you all the contents of his suitcase. Your job is to filter through them all and find the correct pair of tickets.

First, you need to find all valid tickets in his suitcase. Each valid **ticket** has the following characteristics:

* It’s **enclosed** in either **brackets** {} or **square brackets** [].
  + If it contains **mixed opening/closing brackets** (such as {] or [}), **ignore that ticket entirely**
* It contains **any printable ASCII character** inside it
* It contains 2 blocks, enclosed in either **brackets** {} or **square brackets** [], different from the **opening/closing brackets** (examples below):
  + The first will contain the location of their trip, in the format **"CountryCode CityCode"**, with CountryCode being **always** **3 capital latin letters** and CityCode – always **2 capital latin letters**
  + The second will contain the seat number, which consists of **1 capital latin letter** and **1 or 2 digits**

An example valid ticket to Sofia, Bulgaria (BUL SF) looks like this:

|  |
| --- |
| 18dhskf5e{FT\*n\*uk\_[BUL SF]vqre31r/asd[A10]123asd}ad1rsfez |

In **green**, you have the **whole** **ticket** and in **yellow**, the inner blocks with the **location** and **seat** **number** of the **ticket**.

You need to **filter** **only** the **tickets** to the **same** **location** that **Sam** has **told** **you** (see **input**) and **if there are more than 2 valid tickets** for that **location**, take the **2 with the same row number** – the **numeric** **part** of the **seat** **number**.

## Input

* On the **first line** of input, you will receive the **location**, in **the same exact format as in the tickets**
* On the **second line**, you will receive the whole suitcase contents in a single string

## Output

* After **finding** the **correct** **tickets**, print them in the **following** **format**:  
  "You are traveling to {location} on seats {firstSeat} and {secondSeat}."
* The **two** **seats** should be printed in the **same** **row** **they** **were** **found** in the suitcase string

## Constraints

* There will **always** be at least **2 valid tickets**
* There will **never** be more than **2 valid tickets** with the same **row** **number**
* There will **never** be more than **1 valid pair of tickets**

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| BUL SF  1d2ajsd/.{1d9823{BUL SF}10eu2{A11}12das}2fsdf[a2d{BUL SF}12e0dd1rrwg{A11}af/zc,s]d1d0429{d12dasd[LUB SF]123asdAS[A15]fsdf}21ijp3diasd{[BUL SF][B11]}112edasd | You are traveling to BUL SF on seats A11 and B11. | We have only 2 valid tickets here, so we just take their seats and print the results.  The first red ticket is invalid because it uses only one type of brackets.  The second red ticket is valid, but for a different location. |
| BHS PN  1d2ajsd/.{1d9823{BUL SF}10eu2{A11}12das}2fsdf[a2d{BHS PN}12e0dd1rrwg{S35}af/zc,s]d1d0429{d12dasd[BHS PN]123asdAS[A15]fsdf}21ijp3diasd{asda/gjkl.jlk[BHS PN]112easd[T35]pile1s1kartofi}112edasdtarator120asd{asda/gjkl.jlk[BHS PN]112easd[T45]hello??}qr3cf4 67 | You are traveling to BHS PN on seats S35 and T35. | This time we have an invalid ticket – different location and 4 valid tickets, so we take the ones with the same row – 35 (S35 and T35). |
| BHS PN  dsgbvcmv,[123asd{ BHS PN}asdasd{D48}zxcwqc]12dzsdaads2[asd23rf{BHS PN}ghv,/jkm,.n{F48}mbn,buhyj ergd][werg3 54atz {BHS PN}C 3ZSR VG{A1}dg 4a 34 zgdsgf]g45ag | You are traveling to BHS PN on seats F48 and A1. | Ticket [123asd{ BHS PN}asdasd{D48}zxcwqc] is invalid because of the extra whitespace in one of the inner blocks, so we take the other 2 valid tickets. |