**Problem Name:** Check if words Equals Summation of Two Words

**Topics:** String

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**: The **letter value** of a letter is its position in the alphabet **starting from 0** (i.e. 'a' -> 0, 'b' -> 1, 'c' -> 2, etc.).

The **numerical value** of some string of lowercase English letters s is the **concatenation** of the **letter values** of each letter in s, which is then **converted** into an integer.

* For example, if s = "acb", we concatenate each letter's letter value, resulting in "021". After converting it, we get 21.

You are given three strings firstWord, secondWord, and targetWord, each consisting of lowercase English letters 'a' through 'j' **inclusive**.

Return true *if the****summation****of the****numerical values****of*firstWord*and*secondWord*equals the****numerical value****of*targetWord*, or*false*otherwise.*

**Input Format:**

First line of the input contain string first\_word

Second line of the input contain string second\_word

Third line of the input contain string target\_word

Ex:

acb

cba

cdb

**Output Format:** Print Boolean according to problem given.

Ex:

1

**Constraints:**

* 1 <= firstWord.length, secondWord.length, targetWord.length <= 8
* firstWord, secondWord, and targetWord consist of lowercase English letters from 'a' to 'j' **inclusive**.

**Examples:**

**Input:** firstWord = "acb", secondWord = "cba", targetWord = "cdb"

**Output:** true

**Explanation:**

The numerical value of firstWord is "acb" -> "021" -> 21.

The numerical value of secondWord is "cba" -> "210" -> 210.

The numerical value of targetWord is "cdb" -> "231" -> 231.

We return true because 21 + 210 == 231.

**Brute force Solution:**

**Explanation:** Create three vectors for storing values of string as an array then iterate with first, second and target string. Then convert this integer array to the value of that string by multiplying with its 10th power like 123 = 1\*100 + 2\*10 + 3\*1 and last compare whether first + second value is equal to a target value or not

**Code:**

#include <bits/stdc++.h>

using namespace std;

bool isSumEqual(string firstWord, string secondWord, string targetWord) {

    std::vector<int> v1;

    std::vector<int> v2;

    std::vector<int> v3;

    long long int  sum1 = 0;

    long long int sum2 = 0;

    long long int sum3 = 0;

    for (int i = 0 ; i < firstWord.length(); i++) {

        v1.push\_back(firstWord[i] - 'a');

    }

    for (int i = 0 ; i < secondWord.length(); i++) {

        v2.push\_back(secondWord[i] - 'a');

    }

    for (int i = 0 ; i < targetWord.length(); i++) {

        v3.push\_back(targetWord[i] - 'a');

    }

    int j = 0;

    int k = 0;

    int l = 0;

    for (int i = v1.size() - 1; i >= 0; i--) {

        sum1 += v1[i] \* pow(10, j);

        j++;

    }

    for (int i = v2.size() - 1; i >= 0; i--) {

        sum2 += v2[i] \* pow(10, k);

        k++;

    }

    for (int i = v3.size() - 1; i >= 0; i--) {

        sum3 += v3[i] \* pow(10, l);

        l++;

    }

    if (sum1 + sum2 == sum3)

        return true;

    else

        return false;

}

int main() {

    string first, second, target;

    cin>>first>>second>>target;

    cout<<isSumEqual(first,second,target);

    return 0;

}

**Time Complexity**: O(N)

**Space Complexity:** O(N)

**Optimized Solution:**

**Explanation:** Iterate with each string and save its value with proper power at the same step without saving in any other array. Last compare all three string values.

**Code:**

#include <bits/stdc++.h>

using namespace std;

bool isSumEqual(string a, string b, string c) {

    int f1=0,f2=0,f3=0;

    int a1=a.size(),b1=b.size(),c1=c.size();

    for(int i=0;i<a1;i++)

    {

        f1 \*= 10;

        f1 += (a[i]-'a');

    }

    for(int i=0;i<b1;i++)

    {

        f2 \*= 10;

        f2 += (b[i]-'a');

    }

    for(int i=0;i<c1;i++)

    {

        f3 \*= 10;

        f3 += (c[i]-'a');

    }

    return (f1+f2==f3);

}

int main() {

    string first, second, target;

    cin>>first>>second>>target;

    cout<<isSumEqual(first,second,target);

    return 0;

}

**Time Complexity**: O(N)

**Space Complexity:** O(1)