

# Common Child



A string is said to be a child of a another string if it can be formed by deleting 0 or more characters from the other string. Given two strings of equal length, what's the longest string that can be constructed such that it is a child of both?

For example, `ABCD` and `ABDC` have two children with maximum length 3, `ABC` and `ABD`. They can be formed by eliminating either the `D` or `C` from both strings. Note that we will not consider `ABCD` as a common child because we can't rearrange characters and `ABCD`  $\neq$  `ABDC`.

## Function Description

Complete the `commonChild` function in the editor below. It should return the longest string which is a common child of the input strings.

`commonChild` has the following parameter(s):

- `s1, s2`: two equal length strings

## Input Format

There is one line with two space-separated strings, `s1` and `s2`.

## Constraints

- $1 \leq |s1|, |s2| \leq 5000$
- All characters are upper case in the range `ascii[A-Z]`.

## Output Format

Print the length of the longest string `s`, such that `s` is a child of both `s1` and `s2`.

## Sample Input

```
HARRY
SALLY
```

## Sample Output

```
2
```

## Explanation

The longest string that can be formed by deleting zero or more characters from `HARRY` and `SALLY` is `AY`, whose length is 2.

## Sample Input 1

```
AA
BB
```

## Sample Output 1

```
0
```

## Explanation 1

`AA` and `BB` have no characters in common and hence the output is 0.

### Sample Input 2

```
SHINCHAN  
NOHARAAA
```

### Sample Output 2

```
3
```

### Explanation 2

The longest string that can be formed between *SHINCHAN* and *NOHARAAA* while maintaining the order is *NHA*.

### Sample Input 3

```
ABCDEF  
FBDAMN
```

### Sample Output 3

```
2
```

### Explanation 3

*BD* is the longest child of the given strings.