



Hello Igor, [COMPLETE](#) your daily stand up report :-)



# Quest08

Subject

1 Solution

Additional Resources  
(0)

## Quest08

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under `submit directory` and specific file names for each exercise are also provided at the beginning of each exercise under `submit file(s)`.

Quest08	Hiddenp
Submit directory	ex00
Submit file	hiddenp.c

### Control Center



Group formation



Progress



Submitted



Test review



**Finished: approved**



[Go To DoCode](#)



Access:

READ

WRITE

[Go To Gitea](#)

[Keep Working On This Solution](#)

## Description

Write a function named `hidenp` that takes two strings and returns 1

if the first string is hidden in the second one, otherwise returns 0 followed by a newline.

Let `s1` and `s2` be strings. We say that `s1` is hidden in `s2` if it's possible to

find each character from `s1` in `s2`, in the same order as they appear in `s1`.

Also, the empty string is hidden in any string.

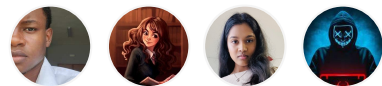
## Function prototype (c)

```
/*
**
** QWASAR.IO -- hidenp
**
** @param {char*} param_1
** @param {char*} param_2
**
** @return {int}
**
*/

int hidenp(char* param_1, char* param_2)
{

}
```

## Also working on the project



[dilan-wi\\_n](#) [urgano\\_va\\_j](#) [sovis\\_r](#) [qurbon\\_ov\\_a](#)

## Just finished



[ahmur\\_ato\\_x](#) [xolxoja\\_y\\_f](#)

Type

Project

Group  
Size

1  
Participant

Review  
system

Test Review (Gandalf)

Difficult

1115



QPoints



mirsalik\_i

n

## Project's Metadata

Project

id: 39

[Learninghat](#) [Gitea](#) Name - Login

Return Value: 1

### Example 01

Input: "abc" && "btarc"

Output:

Return Value: 0



## Example 02

Input: "" && "long string ?ddl"  
Output:  
Return Value: 1

name: [quest08](#)

visible: [True](#)

Quest08	My Split
Submit directory	ex01
Submit file	my_split.c

## Description

We re going to dive into what lies behind the text to column ability used in Excel. For example, say you have a list of 100 names of people who have registered for an event formatted as John Doe but you need to separate first name and last name into their respective columns so that you have John and Doe .

Create a function that splits a string of characters depending on a separator. (sounds crazy....keep reading!)

The second argument is a unique character separator.

The function should return an array which contains a string wrapped between separator.

There cannot be any empty strings in your array.

The string given as an argument of the function won't be modifiable.

All libC functions are forbidden. Except malloc.  
Sizeof is not a function.

## Function prototype (c)

```

/*
**
** QWASAR.IO -- my_split
**
** @param {char*} param_1
** @param {char*} param_2
**
** @return {string_array*}
**
*/

string_array* my_split(char* param_1,
char* param_2)
{

}

```

#### Example 00

Input: "abc def gh-!" && "-"

Output:

Return Value: ["abc def gh", "!" ]

#### Example 01

Input: "abc def gh-!" && " "

Output:

Return Value: ["abc", "def", "gh-!" ]

#### Example 02

Input: "abc def gh!" && "d"

Output:

Return Value: ["abc ", "ef gh!" ]

#### Example 03

Input: "" && ""

Output:

Return Value: []

Quest08	My Strip
Submit directory	ex02
Submit file	my_strip.c

## Description

Write a function that takes a string, and returns another string which contains exactly one space between words, with no spaces or tabs either at the beginning or the end.

A "word" is defined as a part of a string delimited either by spaces/tabs, or by the start/end of the string.

Example 00:

```
Input: "See? It's easy to print the same
thing"
Output: "See? It's easy to print the same
thing"
```

Example 01:

```
Input: " this      time it      will
be    more complex . "
Output: "this time it will be more
complex ."
```

Example 02:

```
Input "No S***  Sherlock..."
Output: "No S*** Sherlock..."
```

Example 03:

Input ""  
Output: ""

### Function prototype (c)

```
/*  
**  
** QWASAR.IO -- my_strip  
**  
** @param {char*} param_1  
**  
** @return {char*}  
**  
*/  
  
char* my_strip(char* param_1)  
{  
  
}
```