We have seen that lists are mutable (they can be changed), and tuples are immutable (they cannot be changed).

Let's try to understand this with an example.

You are given an immutable string, and you want to make changes to it.

Example

```
>>> string = "abracadabra"
```

You can access an index by:

```
>>> print string[5]
a
```

What if you would like to assign a value?

```
>>> string[5] = 'k'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

How would you approach this?

- One solution is to convert the string to a list and then change the value.

Example

```
>>> string = "abracadabra"
>>> l = list(string)
>>> l[5] = 'k'
>>> string = ''.join(l)
>>> print string
abrackdabra
```

- Another approach is to slice the string and join it back.

Example

```
>>> string = string[:5] + "k" + string[6:]
>>> print string
abrackdabra
```

Task

Read a given string, change the character at a given index and then print the modified string.

Function Description

Complete the mutate_string function in the editor below.

mutate_string has the following parameters:

- string string: the string to change
- int position: the index to insert the character at
- string character: the character to insert

Returns

- string: the altered string

Input Format

The first line contains a string, [redacted] .

The next line contains an integer [redacted] , the index location and a string [redacted] , separated by a space.

Sample Input

```
STDIN           Function
-----           --------
abracadabra     s = 'abracadabra'
5 k             position = 5, character = 'k'
```

Sample Output

```
abrackdabra
```

Change Theme Language Python 3

```python
def mutate_string(string, position, character):
    s=list(string)
    s[position]=character
    string=''.join(s)
    return string

if __name__ == '__main__':
    s = input()
    i, c = input().split()
    s_new = mutate_string(s, int(i), c)
    print(s_new)
```

Line: 4 Col: 13

Upload Code as File Test against custom input Run Code Submit Code

You have earned 10.00 points!

You are now 45 points away from the 4th star for your python badge.

59% 175/220

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Compiler Message

Success

Input (stdin) Download

```
abracadabra
5 k
```

Expected Output Download

```
abrackdabra
```

We have seen that lists are mutable (they can be changed), and tuples are immutable (they cannot be changed).

Let's try to understand this with an example.

You are given an immutable string, and you want to make changes to it.

Example

```
>>> string = "abracadabra"
```

You can access an index by:

```
>>> print string[5]
a
```

What if you would like to assign a value?

```
>>> string[5] = 'k'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

How would you approach this?

- One solution is to convert the string to a list and then change the value.

Example

```
>>> string = "abracadabra"
>>> l = list(string)
>>> l[5] = 'k'
>>> string = ''.join(l)
>>> print string
abrackdabra
```

- Another approach is to slice the string and join it back.

Example

```
>>> string = string[:5] + "k" + string[6:]
>>> print string
abrackdabra
```

Task

Read a given string, change the character at a given index and then print the modified string.

Function Description

Complete the mutate_string function in the editor below.

mutate_string has the following parameters:

- string string: the string to change
- int position: the index to insert the character at
- string character: the character to insert

Returns

- string: the altered string

Input Format

The first line contains a string, ***string***.

The next line contains an integer ***position***, the index location and a string ***character***, separated by a space.

Sample Input

```
STDIN           Function
-----           --------
abracadabra     s = 'abracadabra'
5 k             position = 5, character = 'k'
```

Sample Output

```
abrackdabra
```

Change Theme Language Python 3

```python
def mutate_string(string, position, character):
    s=list(string)
    s[position]=character
    string=''.join(s)
    return string

if __name__ == '__main__':
    s = input()
    i, c = input().split()
    s_new = mutate_string(s, int(i), c)
    print(s_new)
```

Line: 4 Col: 13

Test against custom input

You have earned 10.00 points!

You are now 45 points away from the 4th star for your python badge.

59% 175/220

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Compiler Message

```
Success
```

Input (stdin)

```
abracadabra
5 k
```

Expected Output

```
abrackdabra
```