

Oh, behave!

Problem ID: a10p07ohbehave

The wolf, goat and cabbage problem is a puzzle about getting some uncooperative livestock across a river, alive and whole. It dates back to at least the 9th century, and has entered the folklore of several cultures.

In Icelandic, it is best known as formulated in the following time-honored rhyme:

*Hvernig flutt var yfir á,
úlfur, lamb og heypokinn?
Ekkert granda öðru má,
eitt og mann tók báturinn.*

In English, it is stated as follows:

*There is a man by a river, he has a boat and a wolf, a goat and a bag of cabbage.
He wants to move all his belongings over the river but can only take one thing at a time.
When left unsupervised, the goat will eat the cabbage and the wolf will eat the goat.*

Write a program that allows the user to solve the puzzle.

The program should display the board initially like this:

```
You are on the left side  
w g c ~
```

where ~ represents the river.

The user can choose to cross the river with one of the following options:

- w (wolf), g (goat), c (cabbage), n (nothing)

When the user decides to move something over the river, the program needs to check whether the choice is valid, and let the user try again if invalid. If the choice is valid, the program should determine if the puzzle continues. The puzzle stops if the goat eats the cabbage or if the wolf eats the goat, and then the program displays the message:

```
The {consumer} ate the {consumed}.
```

If the user manages to move all the items over the river, the program should display the message:

```
You solved the puzzle!
```

Input

The input begins with one prompt which asks the user which entity they would like to take to the other side of the river with them:

```
What would you like to take over the river? (w/g/c/n):
```

This prompt repeats until either something or someone gets consumed or the puzzle is solved. If at any point the user makes an invalid choice, the program should output:

```
Not a valid choice!
```

and then prompt the user for their input again.

Output

The output consists of two lines initially and for each valid user input.

The first line contains either the string

```
You are on the left side.
```

or

```
You are on the right side.
```

depending on which side the user is on.

The second line contains one string displaying the current state of the puzzle, where each entity is separated by a space.

Sample Input 1

b
w

Sample Output 1

You are on the left side.
w g c ~
What would you like to take over the river? (w/g/c/n):
Not a valid choice!
What would you like to take over the river? (w/g/c/n):
You are on the right side.
g c ~ w
The goat ate the cabbage.

Sample Input 2

g
n
w
n

Sample Output 2

You are on the left side.
w g c ~
What would you like to take over the river? (w/g/c/n):
You are on the right side.
w c ~ g
What would you like to take over the river? (w/g/c/n):
You are on the left side.
w c ~ g
What would you like to take over the river? (w/g/c/n):
You are on the right side.
c ~ g w
What would you like to take over the river? (w/g/c/n):
You are on the left side.
c ~ g w
The wolf ate the goat.

Sample Input 3

g
n
w
g
c
n
g

Sample Output 3

You are on the left side.
w g c ~
What would you like to take over the river? (w/g/c/n):
You are on the right side.
w c ~ g
What would you like to take over the river? (w/g/c/n):
You are on the left side.
w c ~ g
What would you like to take over the river? (w/g/c/n):
You are on the right side.
c ~ g w
What would you like to take over the river? (w/g/c/n):
You are on the left side.
c g ~ w
What would you like to take over the river? (w/g/c/n):
You are on the right side.
g ~ w c
What would you like to take over the river? (w/g/c/n):
You are on the left side.
g ~ w c
What would you like to take over the river? (w/g/c/n):
You are on the right side.
~ w c g
You solved the puzzle!