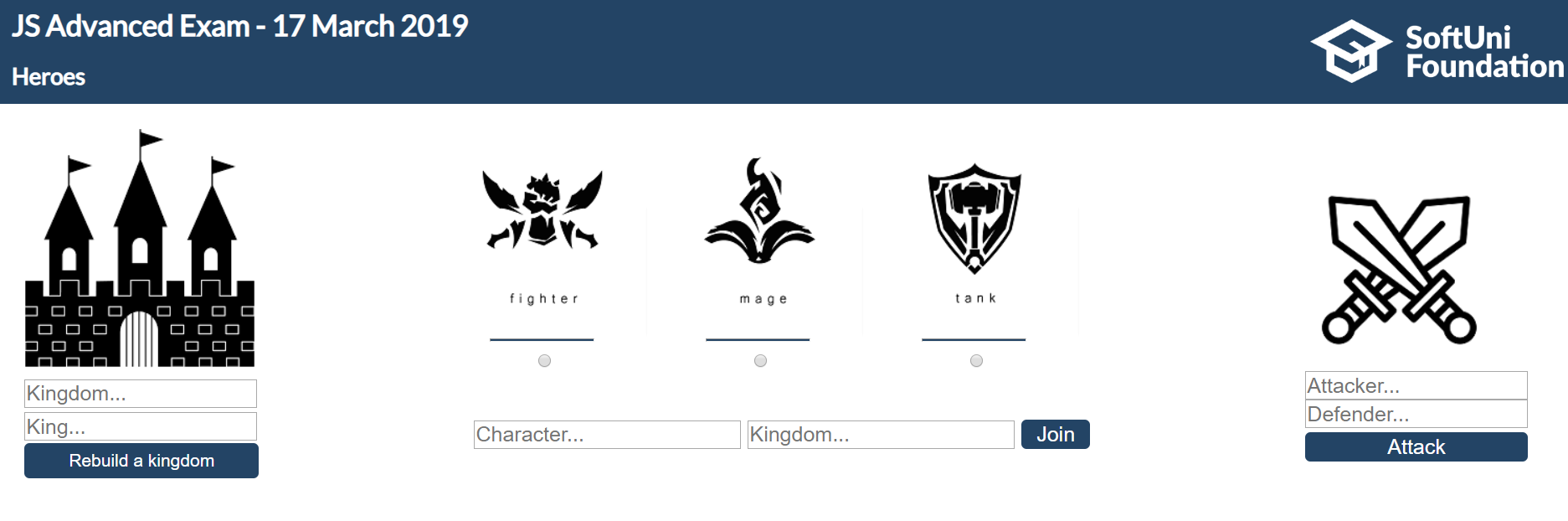
# JS Advanced - Exam: 17.03.2019

Exam problems for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/courses/javascript-advanced). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/Contests/1586/>.

## Problem 4. Heroes

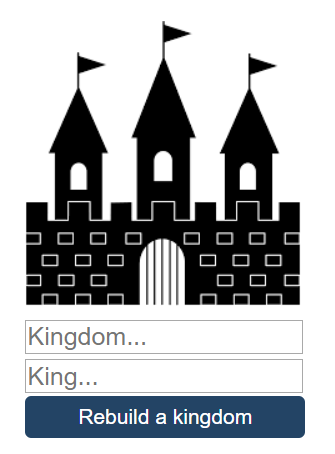
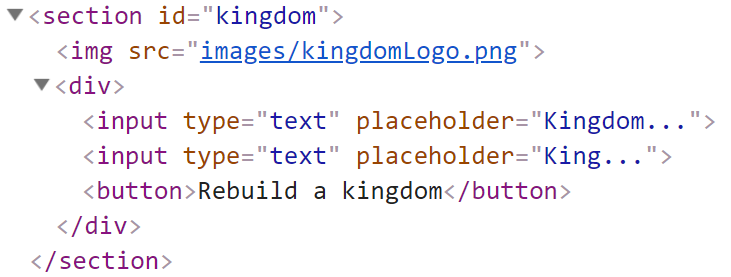
You have **NO permission** to directly change the given HTML file.



This problem should support **three** **types** of functionality:

* **Rebuild a kingdom**
* **Join kingdom**
* **War**

### **Rebuild a kingdom**

* Тhe **king's name** should be a **string** with **length** **greater** or **equal** than **two**.
* The **kingdom's name** can be **one** of the following: **CASTLE**, **DUNGEON**, **FORTRESS**, **INFERNO**, **NECROPOLIS**, **RAMPART**, **STRONGHOLD**, **TOWER** or **CONFLUX. (case insensitive)**  
  

**Note: Every valid kingdom name points to an existing HTML element. By default, every kingdom element is empty and hidden**.



When the [Rebuildakingdom] **button** is **clicked** and **valid input has been received**, **get** that **kingdom** **element** and **create** a few elements within it.

* h1 element for the **kingdom name**
* div element with class="castle"
* h2 element for the **king's name**
* fieldset element for the **army information**

**Each army** has **three** types of characters: "**TANKS**", "**FIGHTERS**" and "**MAGES**".  
Use a p element to hold characters' info in this format: "{type} - {count}"  
And a div element with class="armyOutput" which will hold all **characters'** **names**.

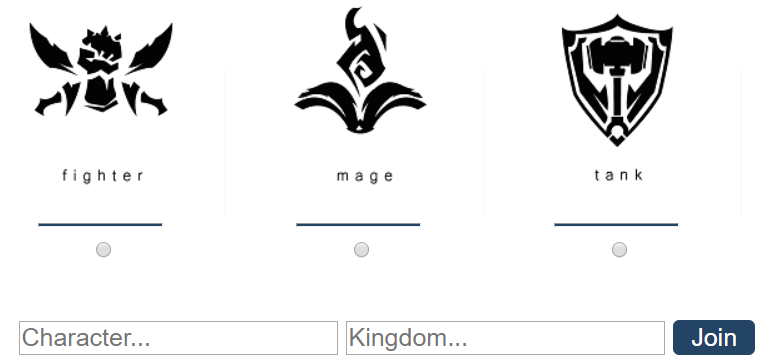


**Note: Kingdom** and **King's names** must be **uppercase**.

#### Example:

### 

### Join kingdom

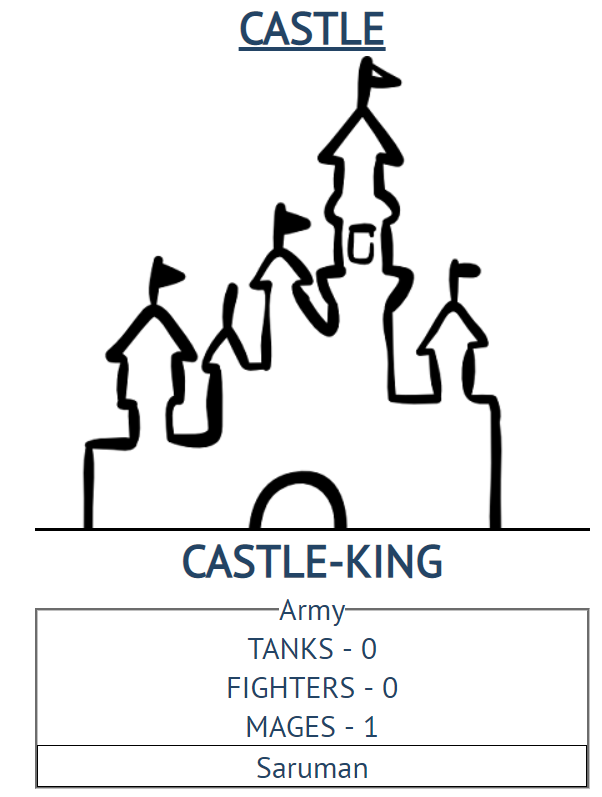


Еvery kingdom can have three types of characters: "**TANKS**", "**FIGHTERS**" and "**MAGES**". This is the moment when a **character joins the kingdom**.

* **Character type** must be **selected**.
* **Character name** must be a **string** with **length** **greater** or **equal** than **two**.
* **The kingdom** which is **joined** needs to be **valid** and that kingdom must have been **rebuilt**. **(case insensitive)**

When the [Join] **button** is **clicked**, **increase** the **count** of that **character** and **add** **their name** to the div which holds all characters' names (div with class="armyOutput"). **Names** must be **separated** with a **single** **space**.

#### Example



### War

In this case, **two valid** and **rebuilt** **kingdoms** start a war with each other. (**The** **kingdom** **names** should be also **case** **insensitive**)  
Here you can see **stats** for each **character**:

* **MAGES** - **70 attack** and **30 defenses**
* **FIGHTERS** - **50 attack** and **50 defenses**
* **TANKS** - **20 attack** and **80 defenses**

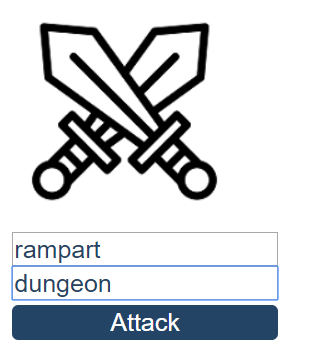
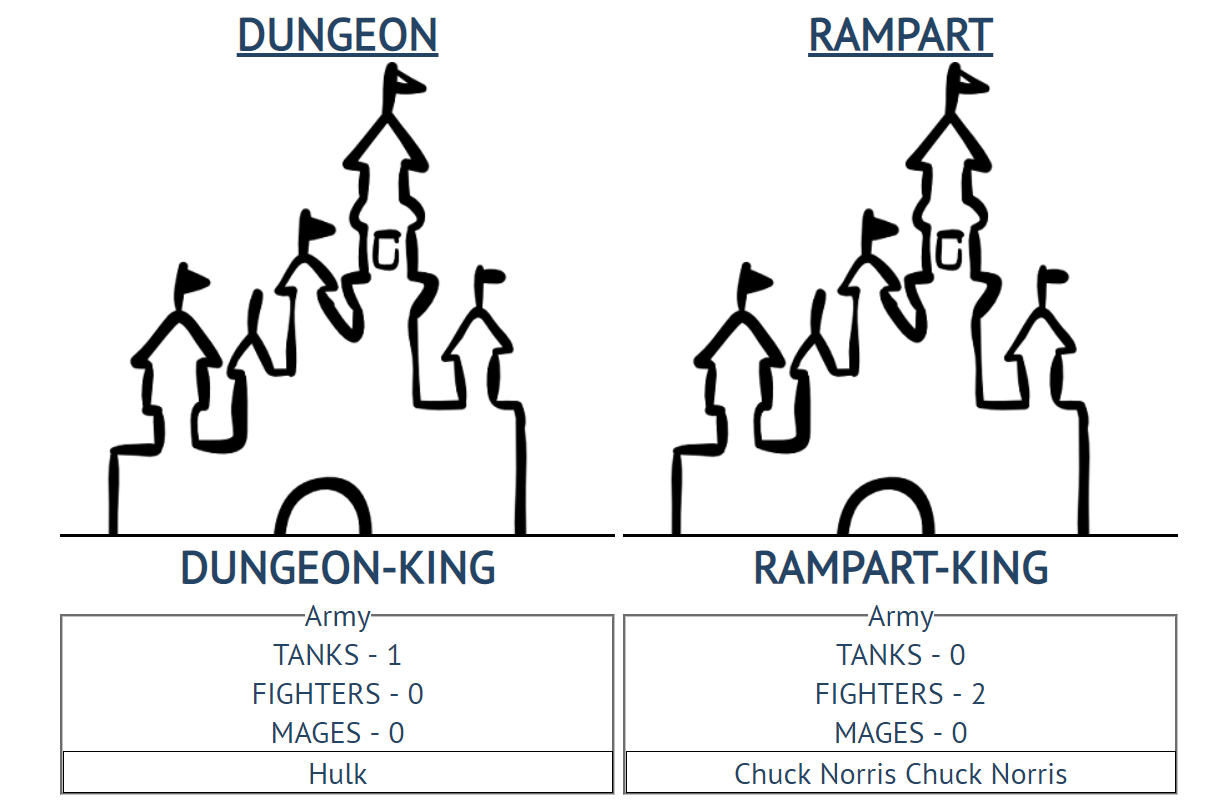
To find out which one will win the war you need to calculate the **attacking** **kingdom's** **total attack points** and the **defense** **kingdom's** **total defense points.**

If the attacking kingdom's points **are more** than the defense kingdom's,the attacking kingdom **wins,** and the defense kingdom has a new king (**The king of the attacking kingdom**).

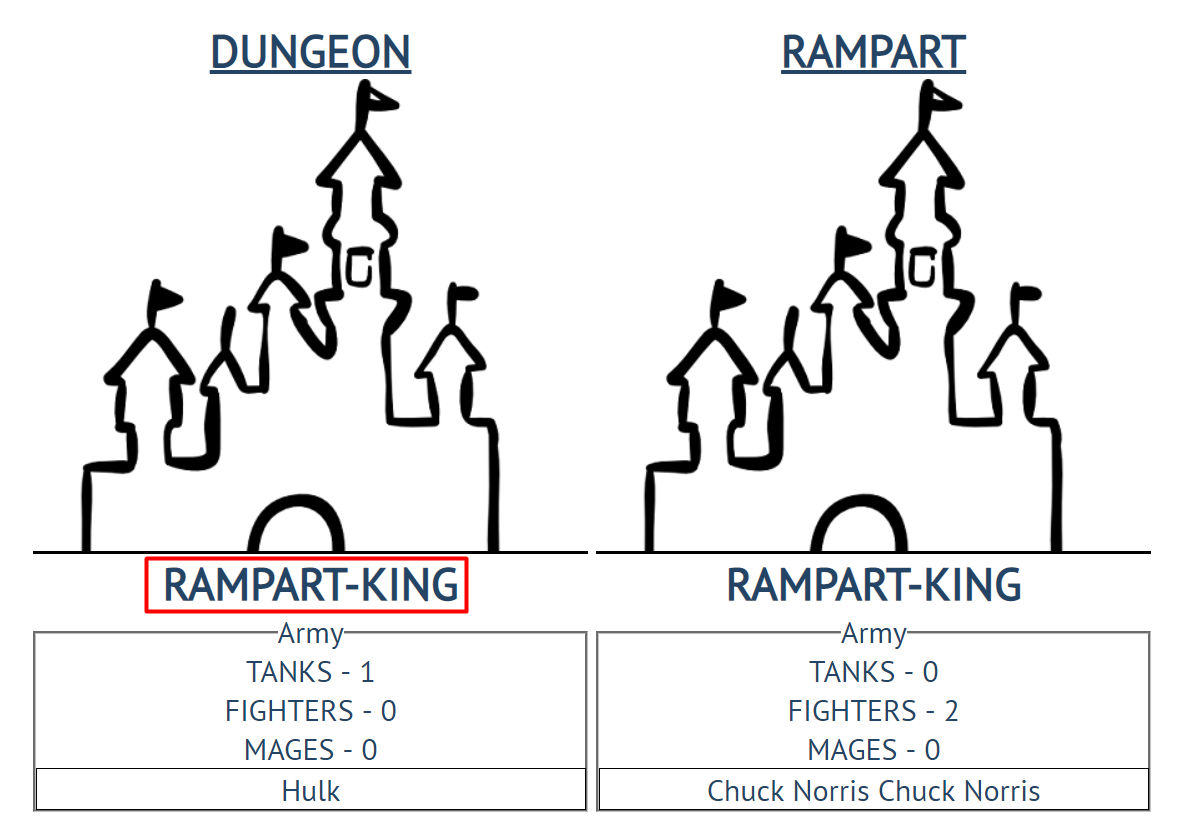
Otherwise, the **defense kingdom** **defends** successfully, and **nothing** **happens**.

#### Example

There are **two** rebuilt kingdoms (**RAMPART** and **DUNGEON**). **RAMPART** have army of **two** **fighters (100 attack points). DUNGEON** have only **one tank** which is (**80** **defense** **points**). **RAMPRT** **attacks** **DUNGEON.**



The result after the war is:



**(RAMPART-KING is the new DUNGEON king, because 100 > 80)**

**Note:** When some **input** **field** receive an **invalid value, clear it.** Otherwisedo **NOT** clearanything!

### Submission

Submit only your **JS code** into a **function**.

