

cc-practice-battle

Table of Contents

- [1. READ README](#)
- [2. TODO Identify yourself](#)
- [3. TODO Nested if statement with numeric input](#)
- [4. TODO Nested if statement with character input](#)

1. READ README

- This file is a practice file for `if...else` statements
- Time: approx. 30 min.
- When you're done with a section move the cursor on the section heading and type `S-<right>` (or `SHIFT+<right-arrow>`).

2. TODO Identify yourself

- replace the placeholder `[yourName]` in the header of this file by your name and save the file (`C-x C-s`).

3. TODO Nested if statement with numeric input

Objective: modify the program [1](#) to take numeric input.

1. Modify [1](#) below to accept input in the form of an integer variable response:
 - When response has the value 1, print its value and "Drink health potion".
 - When response has the value 2, print its value and Resume battle.
 - Parallelize: use only **one** `if...else if` statement.
 - The output should look like this (only one of them):

```
: 1 => Drinking health potion!  
: 2 => Resuming battle!
```

Battle code:

```
int health = 99;  
  
if (health <= 100) {  
    // drink health potion  
    printf("Drinking health potion!\n");  
}  
else {  
    // resume battle  
    printf("Resuming battle!\n");  
}
```

```
Drinking health potion!
```

2. To create an input file, complete the missing parts in the code block [1](#) - or simply write the required input into a file:

```
echo "..." ... battle_input  
cat battle_input
```

— SOLUTION —

```
SRC='./src'  
echo "2" > $SRC/battle_input  
cat $SRC/battle_input
```

```
int response = 0;  
  
scanf("%d", &response);  
  
if (response == 1) {  
    // drink health potion  
    printf("%d => Drinking health potion!\n", response);  
}  
if (response == 2) {  
    // resume battle  
    printf("%d => Resuming battle!\n", response);  
}
```

```
2 => Resuming battle!
```

4. **TODO** Nested if statement with character input

Objective: modify the program to take character input

1. Modify your last program to accept input in the form of a character variable `c_response`
 - When `c_response` has the value `y`, print its value and "Drink health potion".
 - When `c_response` has the value `n`, print its value and Resume battle.
 - Use the conversion specifier `%c` to convert character input
 - The output should look like this (only one of them):

```
: 'y' => Drinking health potion!  
: 'n' => Resuming battle!
```

2. To create an input file, complete the missing parts in the code block [1](#) (or simply write the required input in a file):

```
echo "..." ... battle_input_char  
cat battle_input_char
```

— SOLUTION —

```
SRC='./src'
echo 'n' > $SRC/battle_input_char
cat $SRC/battle_input_char
```

```
char c_response;

scanf("%c", &c_response);

if (c_response == 'y') {
    // drink health potion
    printf("'c' => Drinking health potion!\n",
           c_response);
}
else if (c_response == 'n') {
    // resume battle
    printf("'c' => Resuming battle!\n",
           c_response);
}
```

```
'n' => Resuming battle!
```

Author: [yourName] (pledged)

Created: 2024-04-01 Mon 09:23

[Validate](#)