

## **CS5002 Practice 2 Report**

**200024763**

**2020.10.7**

### **Introduction**

This easy is about the second practice of CS 5002. The main body can be divided into two parts, one is description, another is test and output. Each of them contains three parts, Question1, 2 and 3.

### **Description**

#### *Question1*

The whole design could be divided into three parts.

First part is dealing with reaction time. Outside uses a do-while function to make the code back to begin when the input is NaN or a number within 0.1. Inside asks for the reaction time. To make it comparable, a parseFloat() is used to switch the input to a number, as well as remove alphabet and punctuation. Then an if-else function is used to compare and give an alert.

The second part is dealing with 100/200 and man/woman. Outside using a do-while function to make the code back to begin when input is not expected. Inside asking for 100m/100m type and also using a parseFloat() function to remove punctuations and alphabets. Then using an if-else function to give an alert and back to query if receive the unexpected input such as spaces. Following is the gender block, which is similar to the last one. But as there are too many eligible answers, allGender is used to gathering them. Then a .includes() is used to as the condition.

The third part is dealing with if they broke the world record. Firstly, there is a do-while function with an if-else function inside, which is used to exclude unacceptable input such as NaN. Following is a nested structure block combined with seven if-else functions. Outside is an if-else function to distinguish type === 100 and 200; within it are two if-else functions to distinguish gender; within those two functions, there are four if-else functions to tell if they break the world record and give detail.

#### *Question2*

This project can be divided into two parts. Firstly, there is a function using to tell if the input is part of the alphabet and to avoid 26 (alphabet.length) times, the if-else loop is put outside. Then the function will go through the input until there is a value not belong to the alphabet.

#### *Question3*

The whole program can be divided into two parts.

In the first part, using for loop to make sure user input are numbers 0-40 and input like null

and 0 are excluded. Besides, avoiding the repeating of sysNums and inputNums.

In the second part, an if is used to meet the requirement of 'jackpot'. Then a necked for loop is used to construct the condition of switch foundation. In the end, a switch function is used to distinguish different cases.

## Test and Output

### Question1

For the first part- reaction time. As showing by picture 1, input with lower or upper cases, units behind and punctuations like '0.1s!' is acceptable:

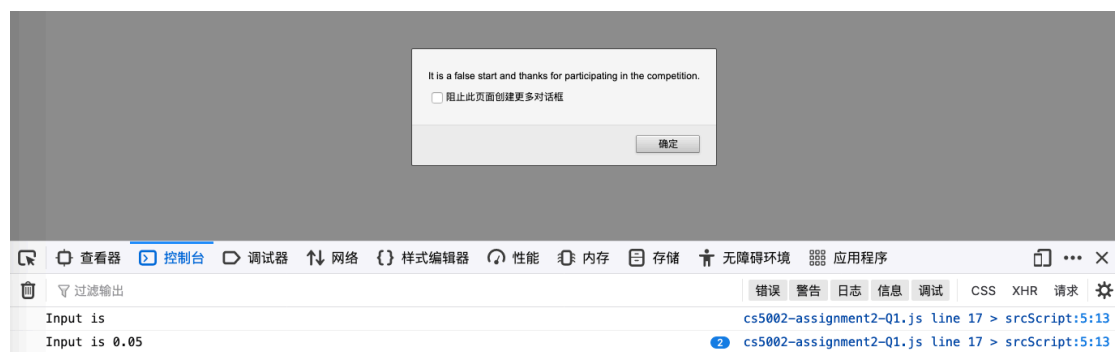


picture 1.1

Then there will be three different common answers: input is NaN like spaces or a punctuation, reaction time  $\geq 0.1s$  and reaction time  $< 0.1s$ . Picture 2, 3 and 4 illustrates that when input is null, it shows 'Please input a right number' then back to the query; if it is 0.05, it shows 'It is a false start and thanks for participating in the competition' then back to the query; if it is 0.1, it goes to next step:



Picture 1.2



Picture 1.3



Picture 1.4

For the second part – distinguishing types and gender. Firstly, there are two different types, which are 100 and 200. Input with lower or upper cases, units behind and punctuations are acceptable such as 100M!. As showing by picture 5:



Picture 1.5

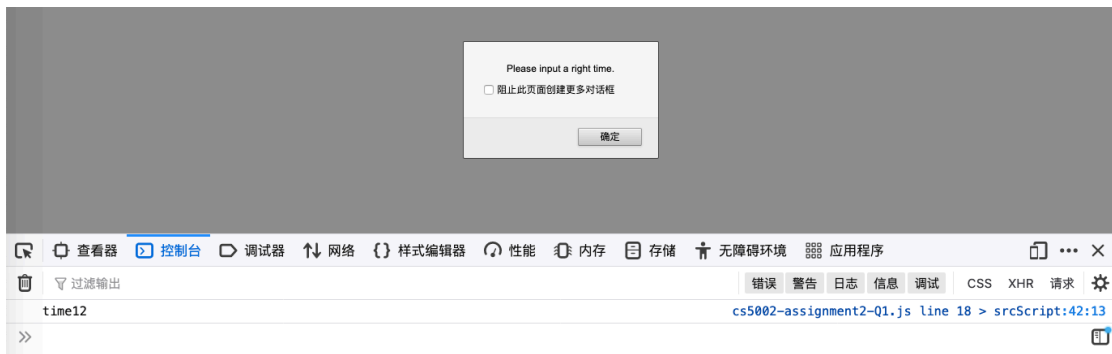
Then if the answer is not acceptable, there will be an alert: Please input the right competition like '\100m\' or '\200m\.' and then back to the query (Picture 6):



Picture 1.6

The same mechanism can be applied to gender validation. What is worth to mention is using replace('\W', '') and toLowerCase() make the input acceptable, such as 'mEn!' and '##womAn's??'.

For the third part – tell if they broke the world record. The input will only keep the digit part inside and the units behind are acceptable but alphabets in front of the number are not, which will back to the query:



Picture 1.7

Then the if-else if is chosen instead of if-else, which is to exclude potential unexpected answer. It will return congratulation and so on when the world record is broken. If not a 'Thanks for ...' will be returned:



Picture 1.8



Picture 1.9

## Question 2

The input such as '!@#' or spaces will be regarded as not a pangrammatic window.



Picture 2.1

If the input is a pangrammatic window, it will alert: 'Text entered is a pangrammatic window.'



Picture 2.2

If not, It will alert: 'Text entered is not a pangrammatic window.'



Picture 2.3

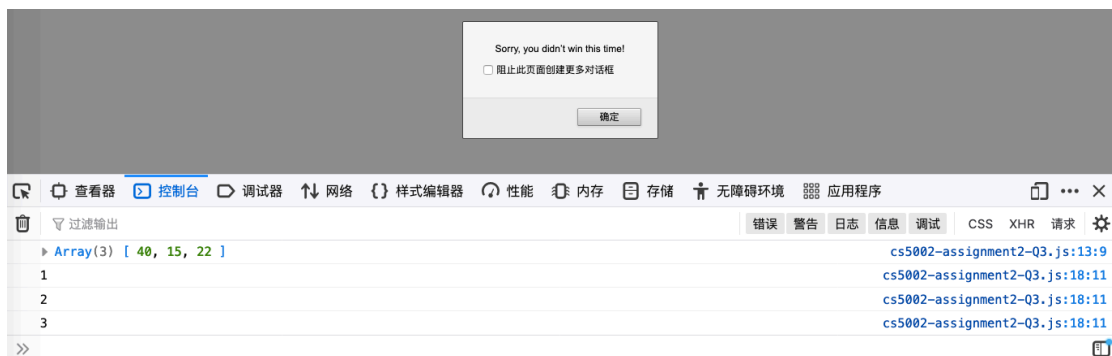
### Question 3

In the first part, inputs are not allowed to contain NaN or input>10 or input < 1. If an input is a float,the .floor() function will transfer it into an integer.



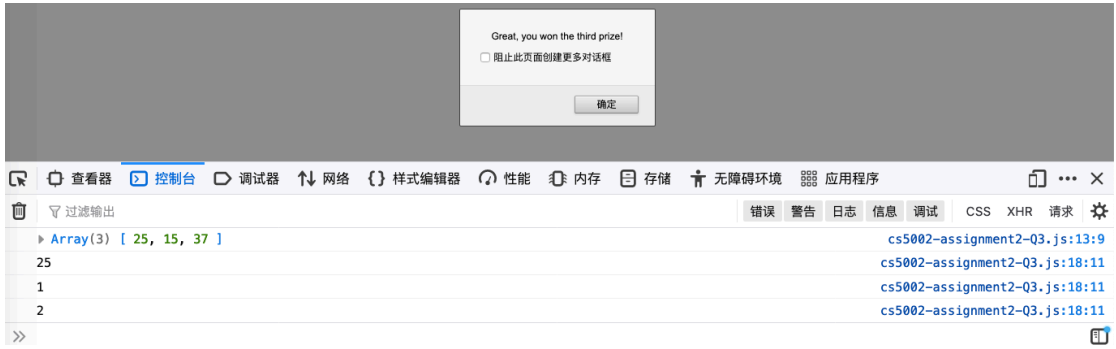
Picture 3.1

In the second part, there will be five different answers. Firstly, is 'did not win':



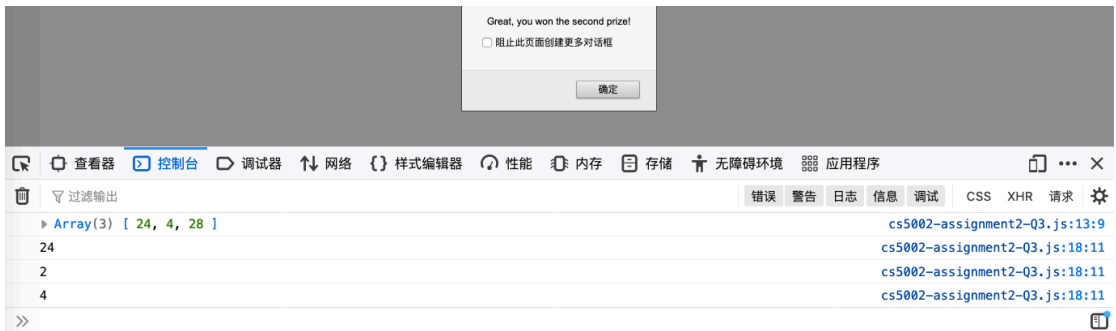
Picture 3.2

The second one is 'the third prize',



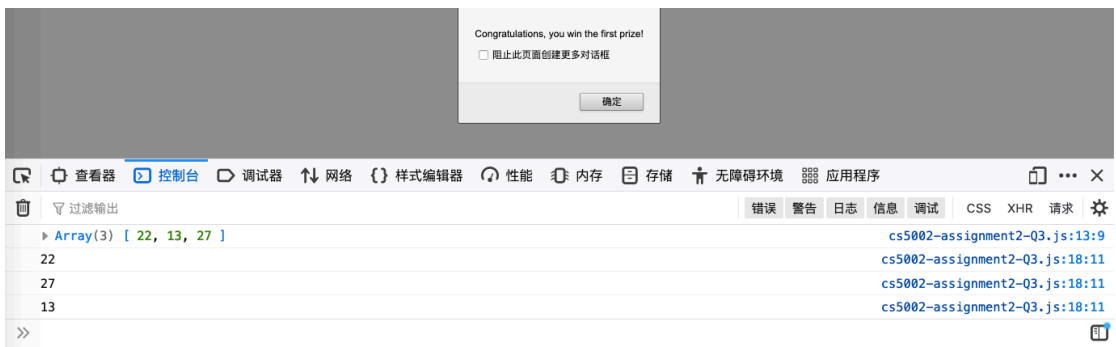
Picture 3.3

The third one is ‘the second prize’ ,



Picture 3.4

The fourth is ‘the first prize’ ,



Picture 3.5

Finally, is ‘the jackpot’ ,



Picture 3.6

## Conclusion

Overall, above has illustrated the process of designing and testing to showing that the three programs meet the needs of the questions. The using of different functions is found to be difficult, especially the switch function of Q3. There will be bugs like losing punctuations and misspelling, also the wrong placing of a statement. The advantage of these programs is thoughtful and well meet the need of the requirements, but the clarity and structure could be improved such as using more effective functions instead of the loops and repeated statement.