★**My firefox version: 92.0.1 (64-bit)**

◆ Introduction

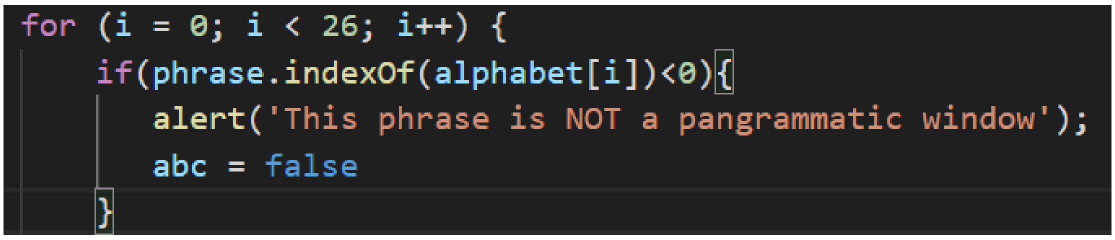
The second practical of Javascript programming gives us an opportunity to practice loops and conditional expressions. Question one requires us to make a programme that analyses the value the users input and tell them whether they have broken the world record or not; In question two we are supposed to design a programme that can analyse whether the input phrases are pangrammatic windows; As for the last question, we are asked to create a lottery game by comparing random integers with input integers and see how many of them are matched. By the end of this practical, students who have accomplished the practical will gain a deeper knowledge and familiarity with the expressions and logics required in this practical.

◆ Question one

For question one, I started off by setting up three variables: event, sex, and time. As we were asked to ignore cases here, I decided to use .toLowerCase() to transfer all the letters that were input into lowercase for later use. Variable sex was a tricky one, we were asked not only to ignore cases but also punctuations. After hours of searching, I finally found a website called Regex 101 (https://regex101.com/), it tells you which Regex to use under different circumstances then tests them on the website to see if we can get the results we are looking for. As shown in my Question1 file, I successfully identified the punctuations and replaced them with '' for later use. The conditional expression I applied into this question was if-else if expression. I had the system first test whether the value for event was 100m or 400m as well as the value for sex; I then designed a follow-up question asking the wind speed for those who answered 100m. The rest were simply setting up conditions according to the world records of different events and genders provided in question one respectively, then had the programme alert whether or not the person had broken the world record based on the results.

◆ Question two

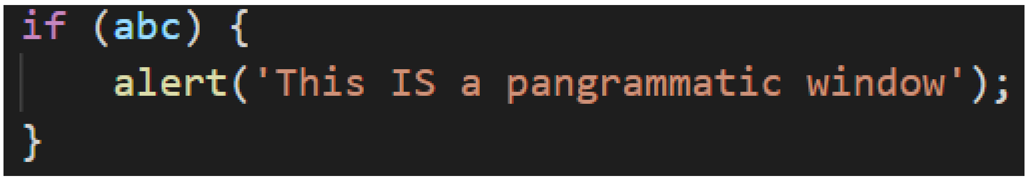
The first part of question two was really easy to me once I figured out how to ignore cases and punctuations. The hard part was to set up a loop and to identify all the English alphabets. I consulted a guy who has been programming for a while and he taught me the whole looping ideas again, which led me to come up with the following code:



```
for (i = 0; i < 26; i++) {
    if(phrase.indexOf(alphabet[i])<0){
        alert('This phrase is NOT a pangrammatic window');
        abc = false
    }
```

The idea is when i < 26, execute if-else expression, that is to find the letter in alphabet[i] and check if it is in phrase's string. It will search from a-z because alphabet is a string of a-z and each time the loop runs, i pluses 1. Since i starts with 0 hence it will start searching from letter 'a' as it is the first letter in alphabet’s string. If a certain letter cannot be found in phrase's string, it returns the value -1, which makes the Boolean in if-else expression true and triggers the statements within. Then the user receives the alert ‘This phrase is NOT a pangrammatic window’.

Notice there is an expression, abc = false. The reason to that is I have set up a Boolean abc = true before entering the loop and if the Boolean from if-else expression does not change abc, it leads to the following statement and alerts ‘This IS a pangrammatic window:

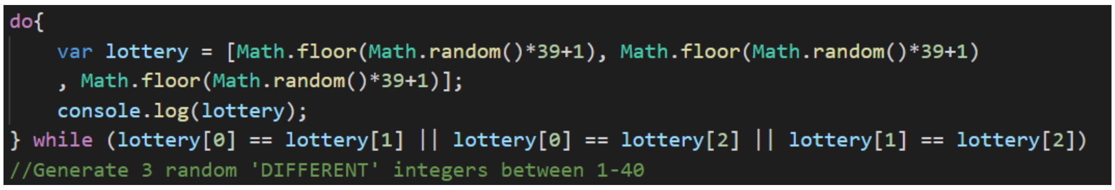


```
if (abc) {
    alert('This IS a pangrammatic window');
}
```

This allows us to test if the phrases input are pangrammatic windows.

Question three

Surprisingly, question three did not take as much time as I expected, despite its complication. With the implementation of array, the question seems to be a lot easier to solve. First, I set up a variable lottery along with do…while loop to prevent any possible numeric repeat. I have also used Math.floor(Math.random()) to make sure the numbers are integers between 1-40.

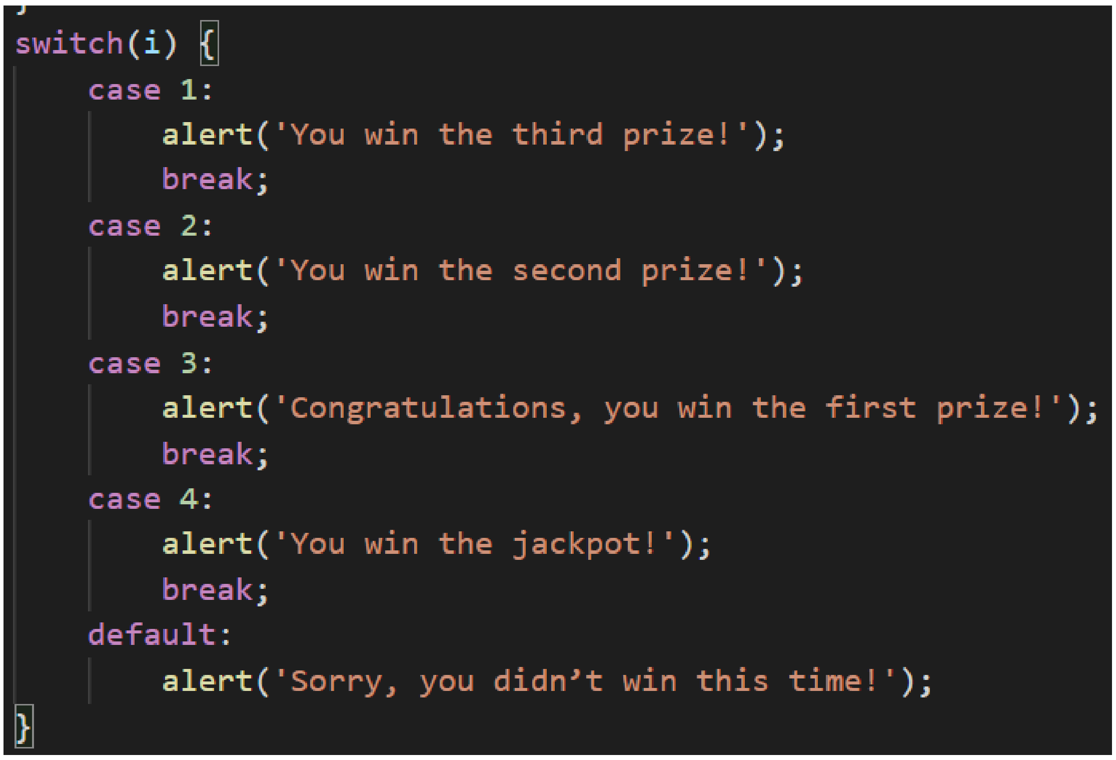


```
do{
    var lottery = [Math.floor(Math.random()*39+1), Math.floor(Math.random()*39+1)
    , Math.floor(Math.random()*39+1)];
    console.log(lottery);
} while (lottery[0] == lottery[1] || lottery[0] == lottery[2] || lottery[1] == lottery[2])
//Generate 3 random 'DIFFERENT' integers between 1-40
```

Next, I set up variable myNum to ask the users to input three non-repeated integers by using a similar apporach. Then it comes to the tricky part; I implemented four if-else expressions to analyse how many numbers of myNum matched lottery. It contained not only Booleans but also and-or logic. I also set up a variable i that equled to 0 before going into the if-else expressions, when any of the test result from the statement returned true, i plused 1.

After a series of testing, the system gets the value of i, 0 means there is not a single number in myNum matches the numbers in lottery; 1 means one number is matched; 2 means two; 3 means three; while 4 is not four, but three numbers with the same exact order as well.

Last but not least, I was going to use if-else expressions but then decided to try the switch-break expression which demonstrated the same effect. Hence I used the code shown in the following picture:



```
switch(i) {
    case 1:
        alert('You win the third prize!');
        break;
    case 2:
        alert('You win the second prize!');
        break;
    case 3:
        alert('Congratulations, you win the first prize!');
        break;
    case 4:
        alert('You win the jackpot!');
        break;
    default:
        alert('Sorry, you didn’t win this time!');
}
```

The code started with switch(i), which would match the value of i into the cases I typed and would switch to default if i did not match any of the cases above. With the codes I used in question three, one can easily have a small lottery game online.

- ## Conclusion

Apparently, this practical is way more challenging than the previous one. However, the sense of accomplishment of coming up with a programme from absolutely nothing is indescribable. The new features I learned from this practical were Regex and array, and despite the time I spent on trying to dig out how to utilise them, I am still satisfied with the outcome as I now am more familiar with when and how to import them into my codes. Other things I knew but was not familiar with were loops and conditional expressions, although they seem pretty easy on the slides, they are easily mistaken in first few tries as one might mistype the location of {} or use the wrong logic that leads to infinite loops. Despite having some minor mental breakdowns while trying to figure out how to solve the practical, I now feel more confident with my coding skills and am gradually growing interest for coding, which is also a good thing since I will feel even more engaged and comfortable with coding.