Problem Solving: Conditional Statements, Iterative Statements, Functions

Relevel
by Unacademy

Why is var outdated in JS?

- Global Scoping
- Hoisting

Global Scoping of var

```
function sayHello() {
    {
        let a = "Rohit";
        var b = "Relevel";
        console.log(a);
    }
    console.log(b); //NO ERROR
}

sayHello();
```

Hoisting of var

```
console.log(a); //NO ERROR, undefined is outputted


var a = 5;
console.log(a); //5
```

Function Hoisting

Hoisting allows functions to be safely used in code before they are declared. Consider the function *add* defined below:

```
1  add(2, 3);
2
3  function add(a, b) {
4    console.log(a + b);
5  }
6  /*
7  The result of the code above is: 5
8  */
```

Here, we might expect the invocation of the function *add* to throw an error since it is being invoked before being defined. However, function hoisting lets us use the function before it is declared in our code and hence the invocation of the *add* function does not throw any error and logs 5 to the console.

Function Expressions are not Hoisted

In the case of functions, only function **declarations** are hoisted–but **not** the function **expressions.** Hence, the below piece of code would throw an error.

```
1  add(2, 3);
2
3  var add = function (a, b) {
4    console.log(a + b);
5  }
6  /*
7  This results in the error:  "Uncaught TypeError: add is not
   a function"
8  */
9
```

Problem - 1A

Write a program to check whether a given number is prime or not.

Example -
num = 17

Output - true

Problem - 1B

Write a program to print all prime numbers less than N

Example -
N = 12

Output -
2
3
5
7
11

Problem - 2A

Write a program to print the digits of a number in any order.

Example -
num = 435

Output -
5
3
4

Core Idea: Divide the number by 10.

On dividing a number by 10, the remainder gives you the last digit of the number.
On dividing a number by 10, the quotient trims the last digit of the number.

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

Output

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1234

Output

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1234

Output

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1234

digit = 4

Output

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1234

digit = 4

Output

```
4
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 123

digit = 4

Output

```
4
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 123

Output

```
4
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 123

digit = 3

Output

```
4
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 123

digit = 3

Output

```
4
3
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 12

digit = 3

Output

```
4
3
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 12

Output

```
4
3
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 12

digit = 2

Output

```
4
3
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 12

digit = 2

Output

```
4
3
2
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1

digit = 2

Output

```
4
3
2
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1

Output

```
4
3
2
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1

digit = 1

Output

```
4
3
2
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 1

digit = 1

Output

```
4
3
2
1
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 0

digit = 1

Output

```
4
3
2
1
```

```
let N = 1234;
while(N > 0) {
    let digit = N % 10;
    console.log(digit);
    N = Math.floor(N / 10);
}
```

N = 0

Output

```
4
3
2
1
```

Problem - 2B

Write a program to calculate the sum of digits of a number.

Example -
num = 435

Output -
12

Problem - 2C

Write a program to check whether a given number is an Armstrong number.
A number is known as an Armstrong number if sum of cube of digits of the number is equal to the number itself.
For e.g. : 153 = 1^3 + 5^3 + 3^3

Example -
num = 370

Output -
true

Problem - 3

Write a program to reverse a number.

Example -
Input - 1234

Output - 4321

Reverse a number

```
let N = 1234;
let reversedNumber = 0;
while(N > 0) {
    let digit = N % 10;
    reversedNumber = reversedNumber * 10 + digit;
    N = Math.floor(N / 10);
}

console.log(reversedNumber);
```

Problem - 3B

Write a program to check whether a given number is palindrome or not.

A number is considered as a palindrome if reads the same from left to right as from right to left

Example -

num = 121

Output - true

num = 17

Output = false

Problem - 3C

Write a program to print all palindrome numbers from 1 to N.

Problem - 4

Create a pattern using ‘*’ as shown below using loops.

*

**

***

****

*****

Problem - 5

Given an even integer, num. Output a pair of numbers (a, b) such that a + b = num and both a and b are prime numbers.

In case multiple pairs exist for a num, output ANYONE of them.

Input - 15

Output - 2, 13

Input - 4

Output

Problem - 5A

Write a program to print the fibonacci sequence till less than N.

Fibonacci sequence - 0, 1, 1, 2, 3, 5, 8, ...

Example -

N = 7

Output

0

1

1

2

3

5

Problem - 5B

Write a program to find the Nth Fibonacci number.

Fibonacci sequence - 0, 1, 1, 2, 3, 5, 8, ...

THANK YOU