

Міністерство Освіти України
Державний університет Телекомунікацій

Лабораторна робота 3
Використання функцій в мові JavaScript

Підготував: студент групи ПД-21
Гапай Максим Юрійович
Перевірів: викладач
Яскевич В.О.

Мета роботи:

- закріпити знання про сутність підпрограм, як засобу підвищення ефективності створення програм;
- відпрацювати технологію створення і реалізації підпрограм в мові програмування;
- придбати практичні навички створення програми з функціями;
- придбати практичні навички створення програми, яка має рекурсію.

Завдання 1.

Код програми:

```
function isEven(n) {  
    n = Math.abs(n);  
    if (n === 0) {  
        return true;  
    } else if (n === 1) {  
        return false;  
    } else {  
        return isEven(n - 2);  
    }  
}
```

Результат виконання:

Парність числа. Рекурсія

Дописати у вікні нижче функцію, яка перевіряє задане число на парність.

Оператор % (залишок від ділення) може використовуватися для визначення того, чи є число парним (% 2). А ось ще один спосіб визначення:

- Нуль парний.
- Одиниця непарна.
- У будь-якого числа N парність така ж, як у $N - 2$.

Напишіть рекурсивну функцію *isEven* згідно з цими правилами. Вона повинна приймати число і повертати булеве значення.

```
1 * function isEven(n) {  
2     n = Math.abs(n);  
3     if (n === 0) {  
4         return true;  
5     } else if (n === 1) {  
6         return false;  
7     } else {  
8         return isEven(n - 2);  
9     }  
10 }
```

chrome 13 Перевірити

OK

OK

OK

OK

Завдання 2.

Код програми:

```
function fib(n, a = 0, b = 1) {  
    return (n > 0) ? fib(n - 1, b, a + b) : a;  
}  
/*  
0 1 1 2 3 5 8...
```

```

start: (a, b) = (b, a + b)
(a, b) = (1, 0 + 1) => (a, b) = (1, 1)
(a, b) = (1, 1 + 1) => (a, b) = (1, 2)
(a, b) = (2, 1 + 2) => (a, b) = (2, 3)
(a, b) = (3, 2 + 3) => (a, b) = (3, 5)
(a, b) = (5, 5 + 3) => (a, b) = (5, 8)
(a, b) = (8, 8 + 5) => (a, b) = (8, 13)
*/

```

Результат виконання:

Послідовність Фібоначчі. Рекурсія

Дописати у вікні нижче функцію, яка повертає n -е число Фібоначчі.

Послідовність чисел Фібоначчі обчислюється за формулою $F(n) = F(n-1) + F(n-2)$. У ній кожне наступне число обчислюється як сума двох попередніх. Перші два числа рівні 1 і 1. Напишіть функцію $fib(n)$, яка повертає n -е число Фібоначчі. Використайте рекурсію.

- `console.log(fib(3)); // 2`
- `console.log(fib(7)); // 13`
- `console.log(fib(77)); // 5527939700884757`

```

1 function fib(n, a = 0, b = 1) {
2   return (n > 0) ? fib(n - 1, b, a + b) : a;
3 }

```

chrome 13

OK
OK
OK
OK
OK
OK

Завдання 3.

Код програми:

```

var sortedSquares = function(nums) {
  return nums.map(x => x * x).sort((a, b) => a - b);
};

```

Результат виконання:

WEB JS: Лабораторна робота 3. x x x Массив: перебирающие методы x (2) Squares of a Sorted Array - Le x +

← → ↺ leetcode.com/problems/squares-of-a-sorted-array/submissions/ Приостановлена

Переводчик YouTube

LeetCode Explore **Day 1** Problems Mock Contest Discuss Store Premium

Description Solution Discuss (999+) Submissions JavaScript Autocomplete

Success Details >

Runtime: 124 ms, faster than 54.16% of JavaScript online submissions for Squares of a Sorted Array.

Memory Usage: 44.2 MB, less than 99.82% of JavaScript online submissions for Squares of a Sorted Array.

Next challenges: Merge Sorted Array Sort Transformed Array

Show off your acceptance: f t in

Time Submitted	Status	Runtime	Memory	Language
04/01/2021 15:02	Accepted	124 ms	44.2 MB	javascript

```
1 /**
2  * @param {number[]} nums
3  * @return {number[]}
4  */
5 var sortedSquares = function(nums) {
6     return nums.map(x => x * x).sort((a, b) => a - b);
7 };
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

Accepted Runtime: 92 ms

Your input [-4,-1,0,3,10]

Output [0,1,9,16,100] Diff

Expected [0,1,9,16,100]

Console Use Example Testcases Run Code Submit

Завдання 4.

Код програми:

```
var sortArrayByParity = function(A) {
    return A.filter(i => i % 2 === 0).concat(A.filter(i => i % 2 !== 0));
};
```

Результат виконання:

WEB JS: Лабораторна робота 3. x x x Массив: перебирающие методы x (2) Sort Array By Parity - LeetCode x +

← → ↺ leetcode.com/problems/sort-array-by-parity/submissions/ Приостановлена

Переводчик YouTube

LeetCode Explore **Day 1** Problems Mock Contest Discuss Store Premium

Description Solution Discuss (999+) Submissions JavaScript Autocomplete

Success Details >

Runtime: 80 ms, faster than 99.69% of JavaScript online submissions for Sort Array By Parity.

Memory Usage: 41.3 MB, less than 48.20% of JavaScript online submissions for Sort Array By Parity.

Next challenges: First Missing Positive Find Pivot Index Add to Array-Form of Integer

Show off your acceptance: f t in

Time Submitted	Status	Runtime	Memory	Language
04/01/2021 15:16	Accepted	80 ms	41.3 MB	javascript

```
1 /**
2  * @param {number[]} A
3  * @return {number[]}
4  */
5 var sortArrayByParity = function(A) {
6     return A.filter(i => i % 2 === 0).concat(A.filter(i => i % 2 !== 0));
7 };
```

Testcase Run Code Result Debugger

Accepted Runtime: 84 ms

Your input [3,1,2,4]

Output [2,4,3,1] Diff

Expected [2,4,3,1]

Console Use Example Testcases Run Code Submit

Завдання 5.

Код програми:

```
var flipAndInvertImage = function(image) {  
    return image.map(i => i.reverse().map( j => j === 0 ? 1 : 0));  
};
```

Результат виконання:

The screenshot shows a web browser window with the URL `leetcode.com/problems/flipping-an-image/submissions/`. The page displays the submission details for the 'Flipping an Image' problem. The submission is successful, with a runtime of 76 ms and memory usage of 40 MB. The code is in JavaScript and uses the `map` function to flip and invert the image matrix.

Success Details >

Runtime: 76 ms, faster than 97.84% of JavaScript online submissions for Flipping an Image.

Memory Usage: 40 MB, less than 96.06% of JavaScript online submissions for Flipping an Image.

Next challenges:

- Pascal's Triangle
- Minimum Domino Rotations For Equal Row
- Binary Prefix Divisible By 5

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
04/01/2021 16:24	Accepted	76 ms	40 MB	javascript

The code editor shows the following JavaScript code:

```
1 /**  
2  * @param {number[][]} image  
3  * @return {number[][]}  
4  */  
5 var flipAndInvertImage = function(image) {  
6     return image.map(i => i.reverse().map( j => j === 0 ? 1 : 0));  
7 };
```

The test case results show:

Accepted Runtime: 84 ms

Your input: `[[1,1,0],[1,0,1],[0,0,0]]`

Output: `[[1,0,0],[0,1,0],[1,1,1]]` [Diff](#)

Expected: `[[1,0,0],[0,1,0],[1,1,1]]`

Висновок: закріпив знання про сутність підпрограм, як засобу підвищення ефективності створення програм, відпрацював технологію створення і реалізації підпрограм в мові програмування, придбав практичні навички створення програми з функціями, а також практичні навички створення програми, яка має рекурсію.