**Вкладення для циклів**

Якщо ви маєте багатовимірний масив, ви можете використати ту ж логіку, що і в попередній точці проходження до циклу через масив і будь-які підмасиви. Ось приклад:

const arr = [

[1, 2], [3, 4], [5, 6]

];

for (let i = 0; i < arr.length; i++) {

for (let j = 0; j < arr[i].length; j++) {

console.log(arr[i][j]);

}

}

Це виводить кожен піделемент у arr по одному. Зверніть увагу, що для внутрішнього циклу ми перевіряємо .length цього arr[i], оскільки arr[i] і є масивом.

Змініть функцію multiplyAll для того, щоб повернути добуток усіх чисел у підмасивах arr.

Запустити тест (Ctrl + Enter)Скинути весь код

Отримати допомогу

**Tests**

* В очікуванні:multiplyAll([[1], [2], [3]]) повинен повернути 6
* В очікуванні:multiplyAll([[1, 2], [3, 4], [5, 6, 7]]) повинен повернути 5040
* В очікуванні:multiplyAll([[5, 1], [0.2, 4, 0.5], [3, 9]]) повинен повернути 54

function multiplyAll(arr) {

let product = 1;

// Змініть код лише під цим рядком

for (let i = 0; i < arr.length; i++) {

for (let j = 0; j < arr[i].length; j++) {

product = product \* arr[i][j];

}

}

// Змініть код лише над цим рядком

return product;

}

multiplyAll([[1, 2], [3, 4], [5, 6, 7]]);