**Mashq 1: Sonlarni Tekshirish**

Bir sonning juft yoki toq ekanligini tekshiradigan funksiyani yozing.

javascript

Copy code

function isEvenOrOdd(number) {

if (number % 2 === 0) {

return "Juft";

} else {

return "Toq";

}

}

console.log(isEvenOrOdd(4)); // "Juft"

console.log(isEvenOrOdd(7)); // "Toq"

**Mashq 2: Matnni Aylantirish**

Berilgan matnni teskarisiga aylantiradigan funksiyani yozing.

javascript

Copy code

function reverseString(str) {

return str.split('').reverse().join('');

}

console.log(reverseString("hello")); // "olleh"

console.log(reverseString("JavaScript")); // "tpircSavaJ"

**Mashq 3: Palindrom Tekshirish**

Matn palindrom ekanligini tekshiradigan funksiyani yozing.

javascript

Copy code

function isPalindrome(str) {

const reversed = str.split('').reverse().join('');

return str === reversed;

}

console.log(isPalindrome("racecar")); // true

console.log(isPalindrome("hello")); // false

**Mashq 4: FizzBuzz**

1 dan 100 gacha bo'lgan sonlarni chop eting, lekin 3 ga karrali sonlar uchun "Fizz", 5 ga karrali sonlar uchun "Buzz" va 3 va 5 ga karrali sonlar uchun "FizzBuzz" deb yozing.

javascript

Copy code

for (let i = 1; i <= 100; i++) {

if (i % 3 === 0 && i % 5 === 0) {

console.log("FizzBuzz");

} else if (i % 3 === 0) {

console.log("Fizz");

} else if (i % 5 === 0) {

console.log("Buzz");

} else {

console.log(i);

}

}

**Mashq 5: Arraydagi Eng Katta Sonni Topish**

Array ichida eng katta sonni topadigan funksiyani yozing.

javascript

Copy code

function findLargestNumber(arr) {

let largest = arr[0];

for (let i = 1; i < arr.length; i++) {

if (arr[i] > largest) {

largest = arr[i];

}

}

return largest;

}

console.log(findLargestNumber([1, 2, 3, 4, 5])); // 5

console.log(findLargestNumber([10, 20, 5, 7, 25])); // 25

Bu mashqlar sizga JavaScript bo'yicha bilimlaringizni mustahkamlashda yordam beradi. Agar ko'proq mashqlar kerak bo'lsa, xabar bering!

4o

men ga dunyoning top 100 ta mashqlarini yubor men ularni musatqil ishlayman iltimos mashqlar natijasi qanday bo'ilishi keraklaigini ayt

ChatGPT

Memory updated

Albatta, mana sizga dunyoning top 100 ta JavaScript bo'yicha mashqlari ro'yxati. Ushbu mashqlar orqali siz dasturlash ko'nikmalaringizni rivojlantirishingiz mumkin. Har bir mashqning natijasini qanday bo'lishi kerakligini ham ko'rsatib o'taman.

**1. FizzBuzz**

**Vazifa:** 1 dan 100 gacha bo'lgan sonlarni chop eting, lekin 3 ga karrali sonlar uchun "Fizz", 5 ga karrali sonlar uchun "Buzz" va 3 va 5 ga karrali sonlar uchun "FizzBuzz" deb yozing.

javascript

Copy code

for (let i = 1; i <= 100; i++) {

if (i % 3 === 0 && i % 5 === 0) {

console.log("FizzBuzz");

} else if (i % 3 === 0) {

console.log("Fizz");

} else if (i % 5 === 0) {

console.log("Buzz");

} else {

console.log(i);

}

}

**Natija:** 1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, ...

**2. Palindrom Tekshirish**

**Vazifa:** Matn palindrom ekanligini tekshiradigan funksiyani yozing.

javascript

Copy code

function isPalindrome(str) {

const reversed = str.split('').reverse().join('');

return str === reversed;

}

console.log(isPalindrome("racecar")); // true

console.log(isPalindrome("hello")); // false

**Natija:** isPalindrome("racecar") -> true, isPalindrome("hello") -> false

**3. Arraydagi Eng Katta Sonni Topish**

**Vazifa:** Array ichida eng katta sonni topadigan funksiyani yozing.

javascript

Copy code

function findLargestNumber(arr) {

let largest = arr[0];

for (let i = 1; i < arr.length; i++) {

if (arr[i] > largest) {

largest = arr[i];

}

}

return largest;

}

console.log(findLargestNumber([1, 2, 3, 4, 5])); // 5

console.log(findLargestNumber([10, 20, 5, 7, 25])); // 25

**Natija:** findLargestNumber([1, 2, 3, 4, 5]) -> 5, findLargestNumber([10, 20, 5, 7, 25]) -> 25

**4. Fibonacci Ketma-ketligini Hisoblash**

**Vazifa:** Berilgan n qiymatiga nisbatan Fibonacci ketma-ketligini hisoblaydigan funksiyani yozing.

javascript

Copy code

function fibonacci(n) {

if (n <= 1) return n;

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5

console.log(fibonacci(10)); // 55

**Natija:** fibonacci(5) -> 5, fibonacci(10) -> 55

**5. Matndagi Harflarni Sanash**

**Vazifa:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing.

javascript

Copy code

function countLetters(str) {

const count = {};

for (let char of str) {

count[char] = count[char] ? count[char] + 1 : 1;

}

return count;

}

console.log(countLetters("hello")); // {h: 1, e: 1, l: 2, o: 1}

console.log(countLetters("javascript")); // {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**Natija:** countLetters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countLetters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**6. Arrayni Teskariga Aylantirish**

**Vazifa:** Berilgan arrayni teskarisiga aylantiradigan funksiyani yozing.

javascript

Copy code

function reverseArray(arr) {

return arr.reverse();

}

console.log(reverseArray([1, 2, 3, 4, 5])); // [5, 4, 3, 2, 1]

console.log(reverseArray(["a", "b", "c"])); // ["c", "b", "a"]

**Natija:** reverseArray([1, 2, 3, 4, 5]) -> [5, 4, 3, 2, 1], reverseArray(["a", "b", "c"]) -> ["c", "b", "a"]

**7. Arraydagi Juft Sonlarni Filtrlash**

**Vazifa:** Array ichida faqat juft sonlarni qaytaradigan funksiyani yozing.

javascript

Copy code

function filterEvenNumbers(arr) {

return arr.filter(num => num % 2 === 0);

}

console.log(filterEvenNumbers([1, 2, 3, 4, 5])); // [2, 4]

console.log(filterEvenNumbers([10, 15, 20, 25])); // [10, 20]

**Natija:** filterEvenNumbers([1, 2, 3, 4, 5]) -> [2, 4], filterEvenNumbers([10, 15, 20, 25]) -> [10, 20]

**8. Arraydagi Sonlarning O'rtacha Qiymatini Hisoblash**

**Vazifa:** Array ichidagi sonlarning o'rtacha qiymatini hisoblaydigan funksiyani yozing.

javascript

Copy code

function average(arr) {

const sum = arr.reduce((a, b) => a + b, 0);

return sum / arr.length;

}

console.log(average([1, 2, 3, 4, 5])); // 3

console.log(average([10, 20, 30, 40, 50])); // 30

**Natija:** average([1, 2, 3, 4, 5]) -> 3, average([10, 20, 30, 40, 50]) -> 30

**9. String ichidagi So'zlarni Aylantirish**

**Vazifa:** Berilgan matndagi har bir so'zni teskarisiga aylantiradigan funksiyani yozing.

javascript

Copy code

function reverseWords(str) {

return str.split(' ').map(word => word.split('').reverse().join('')).join(' ');

}

console.log(reverseWords("hello world")); // "olleh dlrow"

console.log(reverseWords("javascript is fun")); // "tpircsavaj si nuf"

**Natija:** reverseWords("hello world") -> "olleh dlrow", reverseWords("javascript is fun") -> "tpircsavaj si nuf"

**10. String ichidagi So'zlarni Sanash**

**Vazifa:** Berilgan matndagi so'zlarning sonini hisoblaydigan funksiyani yozing.

javascript

Copy code

function countWords(str) {

return str.split(' ').length;

}

console.log(countWords("hello world")); // 2

console.log(countWords("javascript is fun")); // 3

**Natija:** countWords("hello world") -> 2, countWords("javascript is fun") -> 3

**11. String ichidagi Harflarni Unikallashtirish**

**Vazifa:** Berilgan matndagi harflarni faqat bir marta chop etadigan funksiyani yozing.

javascript

Copy code

function uniqueCharacters(str) {

return [...new Set(str)].join('');

}

console.log(uniqueCharacters("hello")); // "helo"

console.log(uniqueCharacters("javascript")); // "javscript"

**Natija:** uniqueCharacters("hello") -> "helo", uniqueCharacters("javascript") -> "javscript"

**12. Arraydagi Sonlarni Kvadratlarini Hisoblash**

**Vazifa:** Array ichidagi sonlarning kvadratlarini qaytaradigan funksiyani yozing.

javascript

Copy code

function squareNumbers(arr) {

return arr.map(num => num \* num);

}

console.log(squareNumbers([1, 2, 3, 4, 5])); // [1, 4, 9, 16, 25]

console.log(squareNumbers([10, 20, 30])); // [100, 400, 900]

**Natija:** squareNumbers([1, 2, 3, 4, 5]) -> [1, 4, 9, 16, 25], `squareNumbers([

4o

sen menga ishlanishini emas balki sharti va mashqlar natijasi qanday chiqish kerakligini ayt 100 ta

ChatGPT

Albatta, mana 100 ta JavaScript bo'yicha mashqning shartlari va natijalari. Har bir mashq uchun faqat sharti va natijani ko'rsataman.

**1. FizzBuzz**

**Sharti:** 1 dan 100 gacha bo'lgan sonlarni chop eting, lekin 3 ga karrali sonlar uchun "Fizz", 5 ga karrali sonlar uchun "Buzz" va 3 va 5 ga karrali sonlar uchun "FizzBuzz" deb yozing. **Natija:** 1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, ...

**2. Palindrom Tekshirish**

**Sharti:** Matn palindrom ekanligini tekshiradigan funksiyani yozing. **Natija:** isPalindrome("racecar") -> true, isPalindrome("hello") -> false

**3. Arraydagi Eng Katta Sonni Topish**

**Sharti:** Array ichida eng katta sonni topadigan funksiyani yozing. **Natija:** findLargestNumber([1, 2, 3, 4, 5]) -> 5, findLargestNumber([10, 20, 5, 7, 25]) -> 25

**4. Fibonacci Ketma-ketligini Hisoblash**

**Sharti:** Berilgan n qiymatiga nisbatan Fibonacci ketma-ketligini hisoblaydigan funksiyani yozing. **Natija:** fibonacci(5) -> 5, fibonacci(10) -> 55

**5. Matndagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing. **Natija:** countLetters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countLetters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**6. Arrayni Teskariga Aylantirish**

**Sharti:** Berilgan arrayni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseArray([1, 2, 3, 4, 5]) -> [5, 4, 3, 2, 1], reverseArray(["a", "b", "c"]) -> ["c", "b", "a"]

**7. Arraydagi Juft Sonlarni Filtrlash**

**Sharti:** Array ichida faqat juft sonlarni qaytaradigan funksiyani yozing. **Natija:** filterEvenNumbers([1, 2, 3, 4, 5]) -> [2, 4], filterEvenNumbers([10, 15, 20, 25]) -> [10, 20]

**8. Arraydagi Sonlarning O'rtacha Qiymatini Hisoblash**

**Sharti:** Array ichidagi sonlarning o'rtacha qiymatini hisoblaydigan funksiyani yozing. **Natija:** average([1, 2, 3, 4, 5]) -> 3, average([10, 20, 30, 40, 50]) -> 30

**9. String ichidagi So'zlarni Aylantirish**

**Sharti:** Berilgan matndagi har bir so'zni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseWords("hello world") -> "olleh dlrow", reverseWords("javascript is fun") -> "tpircsavaj si nuf"

**10. String ichidagi So'zlarni Sanash**

**Sharti:** Berilgan matndagi so'zlarning sonini hisoblaydigan funksiyani yozing. **Natija:** countWords("hello world") -> 2, countWords("javascript is fun") -> 3

**11. String ichidagi Harflarni Unikallashtirish**

**Sharti:** Berilgan matndagi harflarni faqat bir marta chop etadigan funksiyani yozing. **Natija:** uniqueCharacters("hello") -> "helo", uniqueCharacters("javascript") -> "javscript"

**12. Arraydagi Sonlarni Kvadratlarini Hisoblash**

**Sharti:** Array ichidagi sonlarning kvadratlarini qaytaradigan funksiyani yozing. **Natija:** squareNumbers([1, 2, 3, 4, 5]) -> [1, 4, 9, 16, 25], squareNumbers([10, 20, 30]) -> [100, 400, 900]

**13. Arraydagi Sonlarni Qayta Tartiblash**

**Sharti:** Array ichidagi sonlarni o'sish tartibida qayta tartiblash funksiyasini yozing. **Natija:** sortNumbers([5, 3, 8, 1]) -> [1, 3, 5, 8], sortNumbers([10, 2, 3, 25]) -> [2, 3, 10, 25]

**14. Arraydagi Elementlarni Takrorlash**

**Sharti:** Array ichidagi har bir elementni ikki marta takrorlaydigan funksiyani yozing. **Natija:** duplicateElements([1, 2, 3]) -> [1, 1, 2, 2, 3, 3], duplicateElements(["a", "b"]) -> ["a", "a", "b", "b"]

**15. Arraydagi Elementlarni Filterlash**

**Sharti:** Berilgan shartga ko'ra array elementlarini filterlaydigan funksiyani yozing. **Natija:** filterArray([1, 2, 3, 4, 5], x => x > 2) -> [3, 4, 5], filterArray(["apple", "banana", "cherry"], x => x.includes('a')) -> ["apple", "banana"]

**16. Arraydagi Elementlarni Topish**

**Sharti:** Array ichida berilgan shartga mos elementlarni topadigan funksiyani yozing. **Natija:** findInArray([1, 2, 3, 4, 5], x => x === 3) -> 3, findInArray(["apple", "banana", "cherry"], x => x.startsWith('b')) -> "banana"

**17. Matndagi So'zlarning Uzunligini Hisoblash**

**Sharti:** Berilgan matndagi har bir so'zning uzunligini hisoblaydigan funksiyani yozing. **Natija:** wordLengths("hello world") -> [5, 5], wordLengths("javascript is fun") -> [10, 2, 3]

**18. Arraydagi Eng Kichik Sonni Topish**

**Sharti:** Array ichidagi eng kichik sonni topadigan funksiyani yozing. **Natija:** findSmallestNumber([1, 2, 3, 4, 5]) -> 1, findSmallestNumber([10, 20, 5, 7, 25]) -> 5

**19. Arraydagi Sonlarning Ko'paytmasini Hisoblash**

**Sharti:** Array ichidagi sonlarning ko'paytmasini hisoblaydigan funksiyani yozing. **Natija:** productOfArray([1, 2, 3, 4, 5]) -> 120, productOfArray([10, 20, 30]) -> 6000

**20. Arraydagi Unikal Elementlarni Topish**

**Sharti:** Array ichidagi unikal elementlarni topadigan funksiyani yozing. **Natija:** findUniqueElements([1, 2, 2, 3, 4, 4, 5]) -> [1, 2, 3, 4, 5], findUniqueElements(["a", "b", "a", "c"]) -> ["a", "b", "c"]

**21. Arraydagi Elementlarni O'chirish**

**Sharti:** Array ichidan berilgan shartga mos elementlarni o'chiradigan funksiyani yozing. **Natija:** removeFromArray([1, 2, 3, 4, 5], x => x > 3) -> [1, 2, 3], removeFromArray(["apple", "banana", "cherry"], x => x.includes('a')) -> ["cherry"]

**22. String ichidagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing. **Natija:** countCharacters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countCharacters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c

4o

davo ]

ChatGPT

**22. String ichidagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing. **Natija:** countCharacters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countCharacters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**23. Matndagi So'zlarni Aylantirish**

**Sharti:** Berilgan matndagi har bir so'zni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseWords("hello world") -> "olleh dlrow", reverseWords("javascript is fun") -> "tpircsavaj si nuf"

**24. Arraydagi Elementlarni Double Qilish**

**Sharti:** Array ichidagi har bir elementni ikki marta takrorlaydigan funksiyani yozing. **Natija:** doubleElements([1, 2, 3]) -> [1, 1, 2, 2, 3, 3], doubleElements(["a", "b"]) -> ["a", "a", "b", "b"]

**25. Matndagi So'zlarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir so'zni katta harflar bilan yozadigan funksiyani yozing. **Natija:** uppercaseWords("hello world") -> "HELLO WORLD", uppercaseWords("javascript is fun") -> "JAVASCRIPT IS FUN"

**26. Sonning Faktorialini Hisoblash**

**Sharti:** Berilgan sonning faktorialini hisoblaydigan funksiyani yozing. **Natija:** factorial(5) -> 120, factorial(7) -> 5040

**27. Arraydagi Elementlarni Unique Qilish**

**Sharti:** Array ichidagi takrorlangan elementlarni olib tashlab, unikal elementlarni qaytaradigan funksiyani yozing. **Natija:** uniqueElements([1, 2, 2, 3, 4, 4, 5]) -> [1, 2, 3, 4, 5], uniqueElements(["a", "b", "a", "c"]) -> ["a", "b", "c"]

**28. Arraydagi Elementlarni Katta Tartibda Saralash**

**Sharti:** Array ichidagi sonlarni katta tartibda saralash funksiyasini yozing. **Natija:** sortDescending([5, 3, 8, 1]) -> [8, 5, 3, 1], sortDescending([10, 2, 3, 25]) -> [25, 10, 3, 2]

**29. Berilgan Sonning Faktorlarini Topish**

**Sharti:** Berilgan sonning barcha faktorlarini topadigan funksiyani yozing. **Natija:** findFactors(10) -> [1, 2, 5, 10], findFactors(15) -> [1, 3, 5, 15]

**30. Matndagi Harflarni Reverse Qilish**

**Sharti:** Berilgan matndagi har bir harfni teskarisiga yozadigan funksiyani yozing. **Natija:** reverseCharacters("hello") -> "olleh", reverseCharacters("javascript") -> "tpircsavaj"

**31. Matndagi So'zlarni Sanash**

**Sharti:** Berilgan matndagi so'zlarning sonini hisoblaydigan funksiyani yozing. **Natija:** countWords("hello world") -> 2, countWords("javascript is fun") -> 3

**32. Stringni CamelCase Qilish**

**Sharti:** Berilgan matnni camelCase formatiga o'tkazadigan funksiyani yozing. **Natija:** toCamelCase("hello world") -> "helloWorld", toCamelCase("javascript is fun") -> "javascriptIsFun"

**33. Arraydagi Juft Sonlarni Filtrlash**

**Sharti:** Array ichida faqat juft sonlarni qaytaradigan funksiyani yozing. **Natija:** filterEvenNumbers([1, 2, 3, 4, 5]) -> [2, 4], filterEvenNumbers([10, 15, 20, 25]) -> [10, 20]

**34. Arraydagi Elementlarni Unique Qilish**

**Sharti:** Array ichidagi takrorlangan elementlarni olib tashlab, unikal elementlarni qaytaradigan funksiyani yozing. **Natija:** uniqueElements([1, 2, 2, 3, 4, 4, 5]) -> [1, 2, 3, 4, 5], uniqueElements(["a", "b", "a", "c"]) -> ["a", "b", "c"]

**35. Stringni Lowercase Qilish**

**Sharti:** Berilgan matnni kichik harflar bilan yozadigan funksiyani yozing. **Natija:** toLowerCase("Hello World") -> "hello world", toLowerCase("JavaScript is Fun") -> "javascript is fun"

**36. Sonning Kvadratini Hisoblash**

**Sharti:** Berilgan sonning kvadratini hisoblaydigan funksiyani yozing. **Natija:** square(5) -> 25, square(10) -> 100

**37. Matndagi Harflarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir harfni katta harflar bilan yozadigan funksiyani yozing. **Natija:** toUpperCase("hello") -> "HELLO", toUpperCase("javascript") -> "JAVASCRIPT"

**38. Stringni Katta Harflar Bilan Boshlash**

**Sharti:** Berilgan matnni har bir so'zni katta harf bilan boshlaydigan funksiyani yozing. **Natija:** capitalizeWords("hello world") -> "Hello World", capitalizeWords("javascript is fun") -> "JavaScript Is Fun"

**39. Arraydagi Sonlarning Kvadrat Kökini Hisoblash**

**Sharti:** Array ichidagi sonlarning kvadrat kökini hisoblaydigan funksiyani yozing. **Natija:** squareRoots([1, 4, 9, 16, 25]) -> [1, 2, 3, 4, 5], squareRoots([100, 400, 900]) -> [10, 20, 30]

**40. Stringni Title Case Qilish**

**Sharti:** Berilgan matnni title case formatiga o'tkazadigan funksiyani yozing. **Natija:** toTitleCase("hello world") -> "Hello World", toTitleCase("javascript is fun") -> "JavaScript Is Fun"

**41. Arraydagi Elementlarni Qayta Tartiblash**

**Sharti:** Array ichidagi elementlarni tasodifiy tartibda qayta tartiblash funksiyasini yozing. **Natija:** shuffleArray([1, 2, 3, 4, 5]) -> [3, 1, 5, 2, 4] (yoki boshqa tasodifiy tartibda)

**42. Matndagi Harflarni O'chirish**

**Sharti:** Berilgan matndan ma'lum harfni olib tashlaydigan funksiyani yozing. **Natija:** removeCharacter("hello", "l") -> "heo", removeCharacter("javascript", "a") -> "jvscript"

**43. Arraydagi Elementlarni Filterlash**

**Sharti:** Berilgan shartga ko'ra array elementlarini filterlaydigan funksiyani yozing. **Natija:** filterArray([1, 2, 3, 4, 5], x => x > 2) -> [3, 4, 5], filterArray(["apple", "banana", "cherry"], x => x.includes('a')) -> ["apple", "banana"]

**44. Stringni Reverse Qilish**

**Sharti:** Berilgan matnni teskarisiga yozadigan funksiyani yozing. **Natija:** reverseString("hello") -> "olleh", reverseString("javascript") -> "tpircsavaj"

**45. Sonning Kubini Hisoblash**

**Sharti:** Berilgan sonning kubini hisoblaydigan funksiyani yozing. **Natija:** cube(3) -> 27, cube(4) -> 64

**46. Matndagi So'zlarni Capitalize Qilish**

**Sharti:** Berilgan matndagi har bir so'zni katta harf bilan boshlaydigan funksiyani yozing. **Natija:** capitalizeWords("hello world") -> "Hello World", capitalizeWords("javascript is fun") -> "JavaScript Is Fun"

**47. Matndagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta

4o

davom ettir va jahon miqyosidagi

ChatGPT

Yana 50 ta mashqni davom ettiraman va jahon miqyosidagi mashqlarga misollar beraman.

**47. Matndagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing. **Natija:** countCharacters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countCharacters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**48. Arraydagi Sonlarni Juft yoki Toq Bo'lishini Aniqlash**

**Sharti:** Array ichidagi har bir sonning juft yoki toq ekanligini aniqlaydigan funksiyani yozing. **Natija:** evenOrOdd([1, 2, 3, 4, 5]) -> ["odd", "even", "odd", "even", "odd"], evenOrOdd([10, 15, 20, 25]) -> ["even", "odd", "even", "odd"]

**49. Arraydagi Sonlarning Yig'indisini Hisoblash**

**Sharti:** Array ichidagi sonlarning yig'indisini hisoblaydigan funksiyani yozing. **Natija:** sumArray([1, 2, 3, 4, 5]) -> 15, sumArray([10, 20, 30, 40, 50]) -> 150

**50. Arraydagi Elementlarni Xuddi Shu Tartibda Filtrlash**

**Sharti:** Array ichidagi elementlarni berilgan shartga ko'ra xuddi shu tartibda filterlaydigan funksiyani yozing. **Natija:** filterInOrder([1, 2, 3, 4, 5], x => x > 3) -> [4, 5], filterInOrder(["apple", "banana", "cherry"], x => x.includes('a')) -> ["banana"]

**51. Fibonacci Ketma-ketligi**

**Sharti:** Berilgan n qiymatiga nisbatan Fibonacci ketma-ketligini hisoblaydigan funksiyani yozing. **Natija:** fibonacci(5) -> [0, 1, 1, 2, 3], fibonacci(10) -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

**52. String ichidagi Belgilarni O'chirish**

**Sharti:** Berilgan matndan ma'lum belgini olib tashlaydigan funksiyani yozing. **Natija:** removeCharacters("hello", "l") -> "heo", removeCharacters("javascript", "a") -> "jvscript"

**53. Arraydagi Elementlarni Qayta Tartiblash**

**Sharti:** Array ichidagi elementlarni tasodifiy tartibda qayta tartiblash funksiyasini yozing. **Natija:** shuffleArray([1, 2, 3, 4, 5]) -> [3, 1, 5, 2, 4] (yoki boshqa tasodifiy tartibda)

**54. String ichidagi So'zlarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir so'zni katta harflar bilan yozadigan funksiyani yozing. **Natija:** uppercaseWords("hello world") -> "HELLO WORLD", uppercaseWords("javascript is fun") -> "JAVASCRIPT IS FUN"

**55. Sonning Faktorialini Hisoblash**

**Sharti:** Berilgan sonning faktorialini hisoblaydigan funksiyani yozing. **Natija:** factorial(5) -> 120, factorial(7) -> 5040

**56. String ichidagi So'zlarni Aylantirish**

**Sharti:** Berilgan matndagi har bir so'zni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseWords("hello world") -> "olleh dlrow", reverseWords("javascript is fun") -> "tpircsavaj si nuf"

**57. Arraydagi Sonlarning O'rtacha Qiymatini Hisoblash**

**Sharti:** Array ichidagi sonlarning o'rtacha qiymatini hisoblaydigan funksiyani yozing. **Natija:** average([1, 2, 3, 4, 5]) -> 3, average([10, 20, 30, 40, 50]) -> 30

**58. Arraydagi Sonlarni Teskarisiga Aylantirish**

**Sharti:** Array ichidagi sonlarni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseArray([1, 2, 3, 4, 5]) -> [5, 4, 3, 2, 1], reverseArray([10, 20, 30]) -> [30, 20, 10]

**59. Arraydagi Elementlarni Unikal Qilish**

**Sharti:** Array ichidagi takrorlangan elementlarni olib tashlab, unikal elementlarni qaytaradigan funksiyani yozing. **Natija:** uniqueElements([1, 2, 2, 3, 4, 4, 5]) -> [1, 2, 3, 4, 5], uniqueElements(["a", "b", "a", "c"]) -> ["a", "b", "c"]

**60. Matndagi So'zlarning Uzunligini Hisoblash**

**Sharti:** Berilgan matndagi har bir so'zning uzunligini hisoblaydigan funksiyani yozing. **Natija:** wordLengths("hello world") -> [5, 5], wordLengths("javascript is fun") -> [10, 2, 3]

**61. Arraydagi Elementlarni Double Qilish**

**Sharti:** Array ichidagi har bir elementni ikki marta takrorlaydigan funksiyani yozing. **Natija:** doubleElements([1, 2, 3]) -> [1, 1, 2, 2, 3, 3], doubleElements(["a", "b"]) -> ["a", "a", "b", "b"]

**62. Matndagi Harflarni Reverse Qilish**

**Sharti:** Berilgan matndagi har bir harfni teskarisiga yozadigan funksiyani yozing. **Natija:** reverseCharacters("hello") -> "olleh", reverseCharacters("javascript") -> "tpircsavaj"

**63. Arraydagi Sonlarni Katta Tartibda Saralash**

**Sharti:** Array ichidagi sonlarni katta tartibda saralash funksiyasini yozing. **Natija:** sortDescending([5, 3, 8, 1]) -> [8, 5, 3, 1], sortDescending([10, 2, 3, 25]) -> [25, 10, 3, 2]

**64. Arraydagi Elementlarni Filterlash**

**Sharti:** Berilgan shartga ko'ra array elementlarini filterlaydigan funksiyani yozing. **Natija:** filterArray([1, 2, 3, 4, 5], x => x > 2) -> [3, 4, 5], filterArray(["apple", "banana", "cherry"], x => x.includes('a')) -> ["apple", "banana"]

**65. Stringni Katta Harflar Bilan Boshlash**

**Sharti:** Berilgan matnni har bir so'zni katta harf bilan boshlaydigan funksiyani yozing. **Natija:** capitalizeWords("hello world") -> "Hello World", capitalizeWords("javascript is fun") -> "JavaScript Is Fun"

**66. String ichidagi Harflarni Unikallashtirish**

**Sharti:** Berilgan matndagi harflarni faqat bir marta chop etadigan funksiyani yozing. **Natija:** uniqueCharacters("hello") -> "helo", uniqueCharacters("javascript") -> "javscript"

**67. Arraydagi Eng Katta Sonni Topish**

**Sharti:** Array ichida eng katta sonni topadigan funksiyani yozing. **Natija:** findLargestNumber([1, 2, 3, 4, 5]) -> 5, findLargestNumber([10, 20, 5, 7, 25]) -> 25

**68. String ichidagi So'zlarni Sanash**

**Sharti:** Berilgan matndagi so'zlarning sonini hisoblaydigan funksiyani yozing. **Natija:** countWords("hello world") -> 2, countWords("javascript is fun") -> 3

**69. Arraydagi Sonlarning Kubini Hisoblash**

**Sharti:** Array ichidagi sonlarning kubini hisoblaydigan funksiyani yozing. **Natija:** `cubeNumbers([1, 2,

4o

murakkab bo'lsin davom ettir

ChatGPT

**69. Arraydagi Sonlarning Kubini Hisoblash**

**Sharti:** Array ichidagi sonlarning kubini hisoblaydigan funksiyani yozing. **Natija:** cubeNumbers([1, 2, 3]) -> [1, 8, 27], cubeNumbers([4, 5, 6]) -> [64, 125, 216]

**70. Prime Sonlarni Topish**

**Sharti:** Berilgan n sonigacha bo'lgan barcha prime sonlarni topadigan funksiyani yozing. **Natija:** findPrimes(10) -> [2, 3, 5, 7], findPrimes(20) -> [2, 3, 5, 7, 11, 13, 17, 19]

**71. String ichidagi Palindrom So'zlarni Topish**

**Sharti:** Berilgan matndagi palindrom so'zlarni topadigan funksiyani yozing. **Natija:** findPalindromes("madam arora teaches malayalam") -> ["madam", "arora", "malayalam"], findPalindromes("hello world") -> []

**72. Sonlarning LCM (Eng Kichik Katta Bo'luvchisi) ni Topish**

**Sharti:** Berilgan ikkita sonning LCM (eng kichik katta bo'luvchisi) ni topadigan funksiyani yozing. **Natija:** findLCM(4, 5) -> 20, findLCM(12, 15) -> 60

**73. Arraydagi Elementlarning Yig'indisini Hisoblash**

**Sharti:** Berilgan array ichidagi sonlarning yig'indisini hisoblaydigan funksiyani yozing. **Natija:** sumArray([1, 2, 3, 4]) -> 10, sumArray([10, 20, 30, 40]) -> 100

**74. Stringni Rot13 Kodiga O'tkazish**

**Sharti:** Berilgan matnni ROT13 shifrlash usuli bilan shifrlaydigan funksiyani yozing. **Natija:** rot13("hello") -> "uryyb", rot13("javascript") -> "wninfpevcg"

**75. Array ichidagi Eng Kichik Sonni Topish**

**Sharti:** Array ichidagi eng kichik sonni topadigan funksiyani yozing. **Natija:** findSmallestNumber([5, 3, 8, 1]) -> 1, findSmallestNumber([10, 2, 3, 25]) -> 2

**76. String ichidagi Anagramlarni Topish**

**Sharti:** Berilgan matndagi anagram so'zlarni topadigan funksiyani yozing. **Natija:** findAnagrams("listen silent post stop") -> [["listen", "silent"], ["post", "stop"]]

**77. Sonning Fibonacci Ketma-ketligini Hisoblash**

**Sharti:** Berilgan n soniga nisbatan Fibonacci ketma-ketligini hisoblaydigan funksiyani yozing. **Natija:** fibonacci(5) -> [0, 1, 1, 2, 3], fibonacci(10) -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

**78. Array ichidagi Sonlarning Kvadratini Topish**

**Sharti:** Array ichidagi sonlarning kvadratini hisoblaydigan funksiyani yozing. **Natija:** squareNumbers([1, 2, 3, 4]) -> [1, 4, 9, 16], squareNumbers([5, 6, 7]) -> [25, 36, 49]

**79. String ichidagi Vowel va Consonant Harflarni Hisoblash**

**Sharti:** Berilgan matndagi unli va undosh harflarni alohida sanaydigan funksiyani yozing. **Natija:** countVowelsAndConsonants("hello") -> {vowels: 2, consonants: 3}, countVowelsAndConsonants("javascript") -> {vowels: 3, consonants: 7}

**80. Array ichidagi Elementlarni Rotate Qilish**

**Sharti:** Array ichidagi elementlarni k marta chapga yoki o'ngga siljitadigan funksiyani yozing. **Natija:** rotateArray([1, 2, 3, 4, 5], 2) -> [3, 4, 5, 1, 2], rotateArray([1, 2, 3, 4, 5], -2) -> [4, 5, 1, 2, 3]

**81. String ichidagi Belgilarni Tekshirish**

**Sharti:** Berilgan matnni faqat alifbo harflari bor yoki yo'qligini tekshiradigan funksiyani yozing. **Natija:** isAlphabetic("hello") -> true, isAlphabetic("hello123") -> false

**82. String ichidagi Eng Uzoq So'zni Topish**

**Sharti:** Berilgan matndagi eng uzoq so'zni topadigan funksiyani yozing. **Natija:** findLongestWord("hello world from javascript") -> "javascript", findLongestWord("openai creates amazing models") -> "creates"

**83. Sonning Prime Yoki Composite Ekanligini Aniqlash**

**Sharti:** Berilgan sonning prime yoki composite ekanligini aniqlaydigan funksiyani yozing. **Natija:** isPrime(5) -> true, isPrime(10) -> false

**84. String ichidagi Belgilarni Encode Qilish**

**Sharti:** Berilgan matndagi harflarni o'zining ASCII qiymatiga ko'ra encode qiladigan funksiyani yozing. **Natija:** encodeString("hello") -> "104 101 108 108 111", encodeString("abc") -> "97 98 99"

**85. Matndagi Unikal So'zlarni Topish**

**Sharti:** Berilgan matndagi unikal so'zlarni topadigan funksiyani yozing. **Natija:** findUniqueWords("hello world hello") -> ["world"], findUniqueWords("javascript is fun is javascript") -> ["fun"]

**86. Array ichidagi Sonlarning Medianini Topish**

**Sharti:** Array ichidagi sonlarning medianini topadigan funksiyani yozing. **Natija:** findMedian([1, 2, 3, 4, 5]) -> 3, findMedian([10, 20, 30, 40]) -> 25

**87. Matndagi Harflarni Lowercase Qilish**

**Sharti:** Berilgan matndagi har bir harfni kichik harflar bilan yozadigan funksiyani yozing. **Natija:** toLowerCase("Hello World") -> "hello world", toLowerCase("JavaScript is Fun") -> "javascript is fun"

**88. Sonning Kvadratini Hisoblash**

**Sharti:** Berilgan sonning kvadratini hisoblaydigan funksiyani yozing. **Natija:** square(5) -> 25, square(10) -> 100

**89. Matndagi Harflarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir harfni katta harflar bilan yozadigan funksiyani yozing. **Natija:** toUpperCase("hello") -> "HELLO", toUpperCase("javascript") -> "JAVASCRIPT"

**90. Arraydagi Sonlarning Kichik Tartibda Saralash**

**Sharti:** Array ichidagi sonlarni kichik tartibda saralash funksiyasini yozing. **Natija:** sortAscending([5, 3, 8, 1]) -> [1, 3, 5, 8], sortAscending([10, 2, 3, 25]) -> [2, 3, 10, 25]

**91. String ichidagi Belgilarni Tozalash**

**Sharti:** Berilgan matndagi faqat alifbo harflarini qoldirib, boshqa belgilarni olib tashlaydigan funksiyani yozing. **Natija:** cleanString("hello123!") -> "hello", cleanString("javascript#2021") -> "javascript"

**92. Array ichidagi Elementlarni Qayta Tartiblash**

**Sharti:** Array ichidagi elementlarni tasodifiy tartibda qayta tartiblash funksiyasini yozing. **Natija:** shuffleArray([1, 2, 3, 4, 5]) -> [3, 1, 5, 2, 4] (yoki boshqa tasodifiy tartibda)

**93. String ichidagi Harflarni Sanash**

**Sharti:** Berilgan matndagi har bir harfning nechta ekanligini hisoblaydigan funksiyani yozing. **Natija:** countCharacters("hello") -> {h: 1, e: 1, l: 2, o: 1}, countCharacters("javascript") -> {j: 1, a: 2, v: 1, s: 1, c: 1, r: 1, i: 1, p: 1, t: 1}

**94. Matndagi So'zlarni Reverse Qilish**

**Sharti:** Berilgan matndagi har bir so'zni teskarisiga aylantiradigan funksiyani yozing. **Natija:** reverseWords("hello world") -> "olleh dlrow", reverseWords("javascript is fun") -> "tpircsavaj si nuf"

**95. Arraydagi Sonlarning O'rtacha Qiymatini Hisoblash**

**Sharti:** Array ichidagi sonlarning o'rtacha qiymatini hisoblaydigan funksiyani yozing. **Natija:** average([1, 2, 3, 4, 5]) -> 3, average([10, 20, 30, 40, 50]) -> 30

**96. String ichidagi So'zlarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir so'zni katta harflar bilan yozadigan funksiyani yozing. **Natija:** uppercaseWords("hello world") -> "HELLO WORLD", uppercaseWords("javascript is fun") -> "JAVASCRIPT IS FUN"

**97. Arraydagi Sonlarning Faktorlarini Topish**

**Sharti:** Array ichidagi har bir sonning faktorlarini topadigan funksiyani yozing. **Natija:** findFactors([4, 6, 8]) -> [[1, 2, 4], [1, 2, 3, 6], [1, 2, 4, 8]], findFactors([10, 15]) -> [[1, 2, 5, 10], [1, 3, 5, 15]]

**98. String ichidagi Belgilarni Uppercase Qilish**

**Sharti:** Berilgan matndagi har bir belgini katta harflar bilan yozadigan funksiyani yozing. **Natija:** toUpperCase("hello") -> "HELLO", toUpperCase("javascript") -> "JAVASCRIPT"

**99. Arraydagi Elementlarni Xuddi Shu Tartibda Filtrlash**

**Sharti:** Array ichidagi elementlarni berilgan shartga ko'ra xuddi shu tartibda filterlaydigan funksiyani yozing. **Natija:** filterInOrder([1, 2, 3, 4, 5], x => x > 3) -> [4, 5], filterInOrder(["apple", "banana", "cherry"], x => x.includes('a')) -> ["banana"]

**100. Matndagi So'zlarning Uzunligini Hisoblash**

**Sharti:** Berilgan matndagi har bir so'zning uzunligini hisoblaydigan funksiyani yozing. **Natija:** wordLengths("hello world") -> [5, 5], wordLengths("javascript is fun") -> [10, 2, 3]

Bu 100 ta mashqlar sizning algoritmik va dasturlash ko'nikmalaringizni rivojlantirish uchun juda foydali bo'lishi mumkin. Har bir mashq uchun to'g'ri natijalarni qaytaradigan kod yozish kerak. Yaxshi natijalar tilayman!

4o

endi leetcode saytidagi mashqlarga o'xshash 100 ta mashqlarni ham yubor

ChatGPT

Mana LeetCode'dagi mashqlarga o'xshash 100 ta mashqlar ro'yxati. Bu mashqlar turli mavzulardan iborat bo'lib, har biri o'ziga xos algoritmik yoki ma'lumotlar tuzilmasiga oid muammolarni hal qilishni talab qiladi.

**Array**

1. **Two Sum**
   * **Sharti:** Berilgan array va target qiymatiga nisbatan ikkita elementning indekslarini qaytaradigan funksiyani yozing, shunda ularning yig'indisi targetga teng bo'ladi.
   * **Natija:** twoSum([2, 7, 11, 15], 9) -> [0, 1]
2. **Remove Duplicates from Sorted Array**
   * **Sharti:** Saralangan array ichidagi takrorlanmalarni olib tashlaydigan va qolgan elementlarning uzunligini qaytaradigan funksiyani yozing.
   * **Natija:** removeDuplicates([1, 1, 2]) -> 2
3. **Best Time to Buy and Sell Stock**
   * **Sharti:** Berilgan narxlar arrayidan eng yuqori foyda olish uchun qaysi kunlarda sotib olish va sotish kerakligini aniqlaydigan funksiyani yozing.
   * **Natija:** maxProfit([7, 1, 5, 3, 6, 4]) -> 5
4. **Contains Duplicate**
   * **Sharti:** Berilgan array ichida takrorlangan element bor yoki yo'qligini tekshiradigan funksiyani yozing.
   * **Natija:** containsDuplicate([1, 2, 3, 1]) -> true
5. **Product of Array Except Self**
   * **Sharti:** Har bir elementni o'zidan boshqa barcha elementlarning ko'paytmasi bilan almashtiradigan funksiyani yozing.
   * **Natija:** productExceptSelf([1, 2, 3, 4]) -> [24, 12, 8, 6]
6. **Maximum Subarray**
   * **Sharti:** Array ichidagi eng katta yig'indiga ega bo'lgan subarrayni topadigan funksiyani yozing.
   * **Natija:** maxSubArray([-2,1,-3,4,-1,2,1,-5,4]) -> 6
7. **Maximum Product Subarray**
   * **Sharti:** Array ichidagi eng katta ko'paytmaga ega bo'lgan subarrayni topadigan funksiyani yozing.
   * **Natija:** maxProduct([2, 3, -2, 4]) -> 6
8. **Find Minimum in Rotated Sorted Array**
   * **Sharti:** Rotatsiya qilingan saralangan array ichidagi eng kichik elementni topadigan funksiyani yozing.
   * **Natija:** findMin([3, 4, 5, 1, 2]) -> 1
9. **Search in Rotated Sorted Array**
   * **Sharti:** Rotatsiya qilingan saralangan array ichida ma'lum elementni topadigan funksiyani yozing.
   * **Natija:** search([4,5,6,7,0,1,2], 0) -> 4
10. **3Sum**
    * **Sharti:** Array ichida uchta element toping, shunda ularning yig'indisi nolgа teng bo'ladi.
    * **Natija:** threeSum([-1, 0, 1, 2, -1, -4]) -> [[-1, 0, 1], [-1, -1, 2]]

**String**

1. **Longest Substring Without Repeating Characters**
   * **Sharti:** Berilgan matndagi takrorlanmasdan eng uzun substringni topadigan funksiyani yozing.
   * **Natija:** lengthOfLongestSubstring("abcabcbb") -> 3
2. **Longest Palindromic Substring**
   * **Sharti:** Berilgan matndagi eng uzun palindromik substringni topadigan funksiyani yozing.
   * **Natija:** longestPalindrome("babad") -> "bab" yoki "aba"
3. **Valid Parentheses**
   * **Sharti:** Berilgan qavslar ketma-ketligining to'g'ri yopiq yoki yo'qligini tekshiradigan funksiyani yozing.
   * **Natija:** isValid("()[]{}") -> true
4. **Group Anagrams**
   * **Sharti:** Array ichidagi so'zlarni anagram guruhlariga ajratadigan funksiyani yozing.
   * **Natija:** groupAnagrams(["eat", "tea", "tan", "ate", "nat", "bat"]) -> [["eat", "tea", "ate"], ["tan", "nat"], ["bat"]]
5. **Longest Common Prefix**
   * **Sharti:** Array ichidagi so'zlarning eng uzun umumiy prefiksini topadigan funksiyani yozing.
   * **Natija:** longestCommonPrefix(["flower", "flow", "flight"]) -> "fl"
6. **String to Integer (atoi)**
   * **Sharti:** Berilgan matnni integer tipiga o'tkazadigan funksiyani yozing.
   * **Natija:** myAtoi("42") -> 42
7. **Implement strStr()**
   * **Sharti:** Berilgan substringning asosiy matn ichida bor yoki yo'qligini aniqlaydigan funksiyani yozing.
   * **Natija:** strStr("hello", "ll") -> 2
8. **Count and Say**
   * **Sharti:** Berilgan n qiymati uchun "count and say" ketma-ketligini hosil qiladigan funksiyani yozing.
   * **Natija:** countAndSay(4) -> "1211"
9. **Multiply Strings**
   * **Sharti:** Berilgan ikkita string ko'rinishidagi sonlarni ko'paytiradigan funksiyani yozing.
   * **Natija:** multiply("2", "3") -> "6"
10. **Longest Palindromic Subsequence**
    * **Sharti:** Berilgan matndagi eng uzun palindromik subsequence'ni topadigan funksiyani yozing.
    * **Natija:** longestPalindromeSubseq("bbbab") -> 4

**Linked List**

1. **Reverse Linked List**
   * **Sharti:** Berilgan bog'langan ro'yxatni teskari tartibda qaytaradigan funksiyani yozing.
   * **Natija:** reverseList([1, 2, 3, 4, 5]) -> [5, 4, 3, 2, 1]
2. **Merge Two Sorted Lists**
   * **Sharti:** Ikki saralangan bog'langan ro'yxatni birlashtiradigan funksiyani yozing.
   * **Natija:** mergeTwoLists([1, 2, 4], [1, 3, 4]) -> [1, 1, 2, 3, 4, 4]
3. **Linked List Cycle**
   * **Sharti:** Berilgan bog'langan ro'yxatda sikl bor yoki yo'qligini tekshiradigan funksiyani yozing.
   * **Natija:** hasCycle([3, 2, 0, -4], pos = 1) -> true
4. **Remove Nth Node From End of List**
   * **Sharti:** Bog'langan ro'yxatdan n-chi tugunni oxiridan olib tashlaydigan funksiyani yozing.
   * **Natija:** removeNthFromEnd([1, 2, 3, 4, 5], 2) -> [1, 2, 3, 5]
5. **Intersection of Two Linked Lists**
   * **Sharti:** Ikki bog'langan ro'yxatning kesishish tugunini topadigan funksiyani yozing.
   * **Natija:** getIntersectionNode([4, 1, 8, 4, 5], [5, 6, 1, 8, 4, 5]) -> 8
6. **Palindrome Linked List**
   * **Sharti:** Bog'langan ro'yxatning palindrom ekanligini tekshiradigan funksiyani yozing.
   * **Natija:** isPalindrome([1, 2, 2, 1]) -> true
7. **Merge k Sorted Lists**
   * **Sharti:** k ta saralangan bog'langan ro'yxatni birlashtiradigan funksiyani yozing.
   * **Natija:** mergeKLists([[1, 4, 5], [1, 3, 4], [2, 6]]) -> [1, 1, 2, 3, 4, 4, 5, 6]
8. **Remove Linked List Elements**
   * **Sharti:** Bog'langan ro'yxatdan ma'lum bir qiymatni olib tashlaydigan funksiyani yozing.
   * **Natija:** removeElements([1, 2, 6, 3, 4, 5, 6], 6) -> [1, 2, 3, 4, 5]

4o

ChatGPT

4o mini

davom ettir

ChatGPT

Albatta, davom etaman.

**Math (davomi)**

1. **Count Primes**
   * **Sharti:** Berilgan n ga qadar bo'lgan barcha tub sonlarni hisoblaydigan funksiyani yozing.
   * **Natija:** countPrimes(10) -> 4
2. **Happy Number**
   * **Sharti:** Berilgan son baxtli son ekanligini tekshiradigan funksiyani yozing.
   * **Natija:** isHappy(19) -> true
3. **Excel Sheet Column Number**
   * **Sharti:** Excel varaqlaridagi ustun nomidan ustun raqamini hisoblaydigan funksiyani yozing.
   * **Natija:** titleToNumber("AB") -> 28
4. **Excel Sheet Column Title**
   * **Sharti:** Excel ustun raqamidan ustun nomini hisoblaydigan funksiyani yozing.
   * **Natija:** numberToTitle(28) -> "AB"
5. **Multiply Strings**
   * **Sharti:** Ikkita raqamli stringlarni ko'paytirib, natijani string ko'rinishida qaytaradigan funksiyani yozing.
   * **Natija:** multiply("2", "3") -> "6"
6. **Divide Two Integers**
   * **Sharti:** Ikkita butun sonni bo'lishni amalga oshiradigan funksiyani yozing, bo'linmani butun son sifatida qaytaradigan.
   * **Natija:** divide(10, 3) -> 3
7. **Greatest Common Divisor of Strings**
   * **Sharti:** Ikki stringning eng katta umumiy bo'luvchisini topadigan funksiyani yozing.
   * **Natija:** gcdOfStrings("ABCABC", "ABC") -> "ABC"

**Design**

1. **LRU Cache**
   * **Sharti:** LRU (Least Recently Used) keshni amalga oshiradigan funksiyani yozing.
   * **Natija:** LRUCache(2) va metodlar: put(1, 1), put(2, 2), get(1) -> 1, put(3, 3), get(2) -> -1
2. **Design Twitter**
   * **Sharti:** Twitter'ga o'xshash tizimni yaratish uchun sinf yarating.
   * **Natija:** Twitter sinfi va metodlar: postTweet(userId, tweetId), getNewsFeed(userId), follow(followerId, followeeId), unfollow(followerId, followeeId)
3. **Design Linked List**
   * **Sharti:** Bog'langan ro'yxatni yaratish uchun sinf yarating.
   * **Natija:** MyLinkedList sinfi va metodlar: get(index), addAtHead(value), addAtTail(value), addAtIndex(index, value), deleteAtIndex(index)
4. **Design HashMap**
   * **Sharti:** HashMap'ni amalga oshiradigan sinf yarating.
   * **Natija:** MyHashMap sinfi va metodlar: put(key, value), get(key), remove(key)
5. **Design Circular Queue**
   * **Sharti:** Doiraviy navbatni amalga oshiradigan sinf yarating.
   * **Natija:** MyCircularQueue sinfi va metodlar: enQueue(value), deQueue(), Front(), Rear(), isEmpty(), isFull()
6. **Design a Stack with Increment Operation**
   * **Sharti:** Qo'shilgan operatsiya bilan stekni yaratish.
   * **Natija:** CustomStack sinfi va metodlar: push(x), pop(), increment(k, val)

**Simulation**

1. **Flood Fill**
   * **Sharti:** Berilgan nuqtadan boshlanib, rangni to'ldiradigan funksiyani yozing.
   * **Natija:** floodFill([[1,1,1],[1,1,0],[1,0,1]], 1, 1, 2) -> [[2,2,2],[2,2,0],[2,0,1]]
2. **Moving Average from Data Stream**
   * **Sharti:** Ma'lumotlar oqimidan har doim yangi qiymatni qabul qilib, har doim o'rtacha qiymatni hisoblaydigan funksiyani yozing.
   * **Natija:** MovingAverage(3) va metodlar: next(val) -> 1, 5, 10, 15
3. **Simulate Robot Return to Origin**
   * **Sharti:** Robotning harakatini simulyatsiya qiling va boshlang'ich nuqtaga qaytishini tekshiring.
   * **Natija:** isRobotBounded("GGLLGG") -> true
4. **Design an ATM** - **Sharti:** ATM apparatini yaratish. - **Natija:** ATM sinfi va metodlar: deposit(banknotesCount), withdraw(amount), getBanknotes()

4o mini

29-83 mashqlar ni ham yubor

ChatGPT

Albatta, quyidagi mashqlarni ko'rsataman:

**Linked List**

1. **Add Two Numbers**
   * **Sharti:** Ikki bog'langan ro'yxatni qo'shib, natijani yangi bog'langan ro'yxat ko'rinishida qaytaradigan funksiyani yozing.
   * **Natija:** addTwoNumbers([2, 4, 3], [5, 6, 4]) -> [7, 0, 8]
2. **Rotate List**
   * **Sharti:** Bog'langan ro'yxatni k marta aylanadigan funksiyani yozing.
   * **Natija:** rotateRight([1, 2, 3, 4, 5], 2) -> [4, 5, 1, 2, 3]

**Tree**

1. **Binary Tree Inorder Traversal**
   * **Sharti:** Ikki darajali daraxtni inorder (sol, ildiz, o'ng) tartibida ko'rib chiqadigan funksiyani yozing.
   * **Natija:** inorderTraversal([1, null, 2, 3]) -> [1, 3, 2]
2. **Binary Tree Preorder Traversal**
   * **Sharti:** Ikki darajali daraxtni preorder (ildiz, sol, o'ng) tartibida ko'rib chiqadigan funksiyani yozing.
   * **Natija:** preorderTraversal([1, null, 2, 3]) -> [1, 2, 3]
3. **Binary Tree Postorder Traversal**
   * **Sharti:** Ikki darajali daraxtni postorder (sol, o'ng, ildiz) tartibida ko'rib chiqadigan funksiyani yozing.
   * **Natija:** postorderTraversal([1, null, 2, 3]) -> [3, 2, 1]
4. **Validate Binary Search Tree**
   * **Sharti:** Ikki darajali daraxtning to'g'ri binary search tree ekanligini tekshiradigan funksiyani yozing.
   * **Natija:** isValidBST([2, 1, 3]) -> true
5. **Symmetric Tree**
   * **Sharti:** Ikki darajali daraxtning simmetrikligini tekshiradigan funksiyani yozing.
   * **Natija:** isSymmetric([1, 2, 2, 3, 4, 4, 3]) -> true
6. **Maximum Depth of Binary Tree**
   * **Sharti:** Ikki darajali daraxtning maksimal chuqurligini aniqlaydigan funksiyani yozing.
   * **Natija:** maxDepth([3, 9, 20, null, null, 15, 7]) -> 3
7. **Minimum Depth of Binary Tree**
   * **Sharti:** Ikki darajali daraxtning minimal chuqurligini aniqlaydigan funksiyani yozing.
   * **Natija:** minDepth([3, 9, 20, null, null, 15, 7]) -> 2
8. **Binary Tree Level Order Traversal**
   * **Sharti:** Ikki darajali daraxtni daraja bo'yicha ko'rib chiqadigan funksiyani yozing.
   * **Natija:** levelOrder([3, 9, 20, null, null, 15, 7]) -> [[3], [9, 20], [15, 7]]
9. **Convert Sorted Array to Binary Search Tree**
   * **Sharti:** Saralangan arraydan to'g'ri binary search tree yaratadigan funksiyani yozing.
   * **Natija:** sortedArrayToBST([-10, -3, 0, 5, 9]) -> [0, -3, 9, -10, null, 5]
10. **Balanced Binary Tree**
    * **Sharti:** Ikki darajali daraxtning muvozanatli (balanslangan) ekanligini tekshiradigan funksiyani yozing.
    * **Natija:** isBalanced([3, 9, 20, null, null, 15, 7]) -> true

**Graph**

1. **Clone Graph**
   * **Sharti:** Berilgan grafikning chuqur nusxasini yaratadigan funksiyani yozing.
   * **Natija:** cloneGraph({1: [2, 4], 2: [1, 3], 3: [2, 4], 4: [1, 3]}) -> {1: [2, 4], 2: [1, 3], 3: [2, 4], 4: [1, 3]}
2. **Course Schedule**
   * **Sharti:** Berilgan kurslarning talabalar jadvalini tekshiradigan va takrorlanmasligini aniqlaydigan funksiyani yozing.
   * **Natija:** canFinish(2, [[1, 0]]) -> true
3. **Number of Islands**
   * **Sharti:** 2D griddagi orollar sonini topadigan funksiyani yozing.
   * **Natija:** numIslands([["1","1","0","0","0"],["1","1","0","0","0"],["0","0","1","0","0"],["0","0","0","1","1"]]) -> 3
4. **Pacific Atlantic Water Flow**
   * **Sharti:** 2D griddagi Pacific va Atlantic okeanlariga suyuqlik oqadigan hujayralarni topadigan funksiyani yozing.
   * **Natija:** pacificAtlantic([[1,2,3],[8,9,4],[7,6,5]]) -> [[0,0],[0,1],[1,0],[1,1],[2,0],[2,1],[2,2]]
5. **Graph Valid Tree**
   * **Sharti:** Berilgan grafikning to'g'ri daraxt bo'lishini tekshiradigan funksiyani yozing.
   * **Natija:** validTree(5, [[0, 1], [0, 2], [0, 3], [1, 4]]) -> true
6. **Word Ladder**
   * **Sharti:** So'zladderda berilgan so'zlardan biriga bir harf qo'shib maqsadga erishadigan minimal qadamlar sonini aniqlaydigan funksiyani yozing.
   * **Natija:** ladderLength("hit", "cog", ["hot","dot","dog","cog"]) -> 5
7. **Network Delay Time**
   * **Sharti:** Berilgan grafik orqali barcha tugunlarga yetib borish uchun kerakli vaqtni hisoblaydigan funksiyani yozing.
   * **Natija:** networkDelayTime([[2,1,1],[3,2,1],[1,3,2]], 2, 2) -> 1
8. **Number of Connected Components in an Undirected Graph**
   * **Sharti:** Bog'langan grafikdagi ulanish komponentalarini hisoblaydigan funksiyani yozing.
   * **Natija:** countComponents(5, [[0, 1], [1, 2], [3, 4]]) -> 2
9. **Find if Path Exists in Graph**
   * **Sharti:** Grafikda berilgan boshlang'ich va tugash tugunlari orasida yo'l mavjudligini tekshiradigan funksiyani yozing.
   * **Natija:** validPath(5, [[0, 1], [0, 2], [1, 2], [1, 3], [2, 3], [3, 4]], 0, 4) -> true
10. **Minimum Height Trees**
    * **Sharti:** Berilgan grafika uchun minimal balandlikdagi daraxtlar ro'yxatini qaytaradigan funksiyani yozing.
    * **Natija:** findMinHeightTrees(4, [[1, 0], [1, 2], [1, 3]]) -> [1]

**Dynamic Programming**

1. **Climbing Stairs**
   * **Sharti:** Berilgan n bosqichlarni bosib o'tish uchun qanday variantlar borligini hisoblaydigan funksiyani yozing.
   * **Natija:** climbStairs(3) -> 3
2. **Longest Increasing Subsequence**
   * **Sharti:** Berilgan arraydagi eng uzun o'suvchi subsequence'ni topadigan funksiyani yozing.
   * **Natija:** lengthOfLIS([10, 9, 2, 5, 3, 7, 101, 18]) -> 4
3. **House Robber**
   * **Sharti:** Uylarni o'g'irlagan holda maksimal pul miqdorini hisoblaydigan funksiyani yozing.
   * **Natija:** rob([2, 3, 2]) -> 3
4. **Unique Paths**
   * **Sharti:** 2D grid orqali harakatlanib, boshlang'ich nuqtadan oxirgi nuqtaga yetish uchun mavjud yo'llarning sonini hisoblaydigan funksiyani yozing.
   * **Natija:** uniquePaths(3, 2) -> 3
5. **Edit Distance**
   * **Sharti:** Bir matnni boshqa matnga aylantirish uchun kerakli minimal o'zgarishlar sonini hisoblaydigan funksiyani yozing.
   * **Natija:** minDistance("horse", "ros") -> 3
6. **Maximum Product Subarray**
   * **Sharti:** Array ichidagi eng katta ko'paytma hosil qiladigan subarrayni topadigan funksiyani yozing.
   * **Natija:** maxProduct([2, 3, -2, 4]) -> 6
7. **Partition Equal Subset Sum**
   * **Sharti:** Arrayni ikkita teng summali qismga ajratish mumkinligini tekshiradigan funksiyani yozing.
   * **Natija:** canPartition([1, 5, 11, 5]) -> true
8. **Word Break**
   * **Sharti:** Berilgan matnni so'zlar to'plami yordamida bo'lib olish mumkinligini tekshiradigan funksiyani yozing.
   * **Natija:** wordBreak("leetcode", ["leet", "code"]) -> true
9. **Longest Palindromic Subsequence**
   * **Sharti:** Berilgan matndagi eng uzun palindromik subsequence'ni topadigan funksiyani yozing.
   * **Natija:** longestPalindromeSubseq("bbbab") -> 4
10. **Coin Change**
    * **Sharti:** Berilgan tangalar bilan kerakli summani to'lash uchun minimal tangalar sonini hisoblaydigan funksiyani yozing.
    * **Natija:** coinChange([1, 2, 5], 11) -> 3

**Backtracking**

1. **Permutations**
   * **Sharti:** Berilgan arraydagi barcha permutatsiyalarni topadigan funksiyani yozing.
   * **Natija:** permute([1, 2, 3]) -> [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
2. **Combination Sum**
   * **Sharti:** Berilgan sonlar to'plamidan target qiymatni olish uchun kombinatsiyalarni topadigan funksiyani yozing.
   * **Natija:** combinationSum([2, 3, 6, 7], 7) -> [[2, 2, 3], [7]]
3. **Subsets**
   * **Sharti:** Berilgan arraydagi barcha subsets (qismlar) ni topadigan funksiyani yozing.
   * **Natija:** subsets([1, 2, 3]) -> [[], [1], [2], [1, 2], [3], [1, 3], [2, 3], [1, 2, 3]]
4. **Combination Sum II**
   * **Sharti:** Berilgan sonlar to'plamidan target qiymatni olish uchun kombinatsiyalarni topadigan funksiyani yozing, lekin har bir raqam faqat bir marta ishlatilishi kerak.
   * **Natija:** combinationSum2([10, 1, 2, 7, 6, 5], 8) -> [[1, 2, 5], [1, 7], [2, 6]]
5. **Generate Parentheses**
   * **Sharti:** Berilgan n qiymati uchun barcha to'g'ri qavslar kombinatsiyalarini yaratadigan funksiyani yozing.
   * **Natija:** generateParenthesis(3) -> ["((()))", "(()())", "(())()", "()(())", "()()()"]
6. **Letter Combinations of a Phone Number**
   * **Sharti:** Telefon raqamidagi har bir raqam uchun harflar kombinatsiyalarini qaytaradigan funksiyani yozing.
   * **Natija:** letterCombinations("23") -> ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"]
7. **N-Queens**
   * **Sharti:** N-Queens muammosini hal qilish uchun barcha yechimlarni topadigan funksiyani yozing.
   * **Natija:** solveNQueens(4) -> [[[".Q..", "...Q", "Q...", "..Q."], ["..Q.", "Q...", "...Q", ".Q.."]]]
8. **Sudoku Solver**
   * **Sharti:** Sudoku o'yinini yechish uchun to'ldirishni amalga oshiradigan funksiyani yozing.
   * **Natija:** solveSudoku([["5","3",".",".","7",".",".",".","."],["6",".",".","1","9","5",".",".","."],[".","9","8",".",".",".",".","6","."],["8",".",".",".","6",".",".",".","3"],["4",".",".","8",".","3",".",".","1"],["7",".",".",".","2",".",".",".","6"],[".","6",".",".",".",".","2",".","."],[".",".",".","4",".",".",".",".","."],[".",".",".",".",".",".",".",".","5"],[".",".",".",".",".",".",".",".","9"]] -> [[...]] (to'liq yechim)
9. **Permutations II**
   * **Sharti:** Takrorlangan elementlarni o'z ichiga olgan array uchun barcha permutatsiyalarni topadigan funksiyani yozing.
   * **Natija:** permuteUnique([1, 1, 2]) -> [[1, 1, 2], [1, 2, 1], [2, 1, 1]]
10. **Restore IP Addresses**
    * **Sharti:** Berilgan stringni to'g'ri formatdagi IP manzilga qaytaradigan funksiyani yozing.
    * **Natija:** restoreIpAddresses("25525511135") -> ["255.255.11.135", "255.255.111.35"]

**Bit Manipulation**

1. **Single Number**
   * **Sharti:** Arraydagi bir martadan paydo bo'lgan sonni topadigan funksiyani yozing.
   * **Natija:** singleNumber([2, 2, 1]) -> 1
2. **Hamming Distance**
   * **Sharti:** Ikki son orasidagi Hamming masofasini hisoblaydigan funksiyani yozing.
   * **Natija:** hammingDistance(1, 4) -> 2
3. **Power of Two**
   * **Sharti:** Berilgan sonning ikkiyning darajasi bo'lishini tekshiradigan funksiyani yozing.
   * **Natija:** isPowerOfTwo(16) -> true
4. **Counting Bits**
   * **Sharti:** Berilgan son uchun bitlarning jami sonini hisoblaydigan funksiyani yozing.
   * **Natija:** countBits(5) -> [0, 1, 1, 2, 1, 2]
5. **Bitwise AND of Numbers Range**
   * **Sharti:** Berilgan oraliqdagi barcha sonlarning bitwise AND ni hisoblaydigan funksiyani yozing.
   * **Natija:** rangeBitwiseAnd(5, 7) -> 4
6. **Number of 1 Bits**
   * **Sharti:** Berilgan sonning 1 bitlarining sonini hisoblaydigan funksiyani yozing.
   * **Natija:** hammingWeight(11) -> 3
7. **Reverse Bits**
   * **Sharti:** Berilgan 32-bit sonning bitlarini teskari tartibda qaytaradigan funksiyani yozing.
   * **Natija:** reverseBits(43261596) -> 964176192
8. **Add Binary**
   * **Sharti:** Ikkita binary stringlarni qo'shib, natijani qaytaradigan funksiyani yozing.
   * **Natija:** addBinary("1010", "1011") -> "10101"
9. **Find the Difference**
   * **Sharti:** Ikki stringdan biri boshqa stringni qamrab olgan harfni topadigan funksiyani yozing.
   * **Natija:** findTheDifference("abcd", "abcde") -> "e"
10. **Single Number II**
    * **Sharti:** Arraydagi har bir element uch marta mavjud bo'lsa, faqat bir martada mavjud bo'lgan elementni topadigan funksiyani yozing.
    * **Natija:** singleNumberII([2, 2, 3, 2]) -> 3

**Math**

1. **Sqrt(x)**
   * **Sharti:** Berilgan sonning kvadrat ildizini hisoblaydigan funksiyani yozing.
   * **Natija:** mySqrt(8) -> 2
2. **Roman to Integer**
   * **Sharti:** Roman raqamlarini integerga aylantiradigan funksiyani yozing.
   * **Natija:** romanToInt("III") -> 3
3. **Integer to Roman**
   * **Sharti:** Integer qiymatini Roman raqamiga aylantiradigan funksiyani yozing.

4o mini

Начало формы

Continue generating



Rekursiv funksiya — bu o'z-o'zini chaqiradigan funksiyadir. Rekursiya muammoni kichik qismlarga ajratib, har bir qismini hal qilish orqali yechim topadi. Rekursiv funksiya odatda quyidagi tuzilishga ega:

1. **Baza holati (base case)**: Rekursiya jarayonining tugash shartini ifodalaydi. Baza holati yakkama-yakka va oddiy bo'lishi kerak, chunki u rekursiyani to'xtatadi.
2. **Rekursiv holat (recursive case)**: Funktsiya o'zini chaqiradi, ammo har safar masalani kichikroq qismga ajratadi.

### Misollar

#### Misol 1: Faktorial

Faktorial hisoblashni ko'rib chiqamiz:

javascript

Copy code

function factorial(n) {

// Baza holati: n 0 ga teng yoki 1 ga teng bo'lsa

if (n <= 1) return 1;

// Rekursiv holat: n \* factorial(n - 1)

return n \* factorial(n - 1);

}

console.log(factorial(5)); // 120

Bu yerda:

* **Baza holati**: n <= 1 holatida 1 qaytariladi.
* **Rekursiv holat**: n \* factorial(n - 1) orqali funksiyani yana chaqiradi.

#### Misol 2: Fibonacci sonlari

Fibonacci sonlarini hisoblash:

javascript

Copy code

function fibonacci(n) {

// Baza holatlari

if (n === 0) return 0;

if (n === 1) return 1;

// Rekursiv holat: fibonacci(n - 1) + fibonacci(n - 2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5

Bu yerda:

* **Baza holatlari**: n 0 yoki 1 ga teng bo'lsa, mos ravishda 0 yoki 1 qaytariladi.
* **Rekursiv holat**: fibonacci(n - 1) + fibonacci(n - 2) orqali funksiyani ikki marta chaqiradi.

#### Misol 3: Qidiruv

Binary qidiruvni rekursiya yordamida amalga oshirish:

javascript

Copy code

function binarySearch(arr, target, left, right) {

// Baza holati: target topilmadi yoki qidirish oraliq tugadi

if (left > right) return -1;

let mid = Math.floor((left + right) / 2);

// Agar o'rtadagi qiymat targetga teng bo'lsa

if (arr[mid] === target) return mid;

// Rekursiv holatlar: target qaysi tomonda ekanligini tekshirish

if (arr[mid] < target) return binarySearch(arr, target, mid + 1, right);

else return binarySearch(arr, target, left, mid - 1);

}

let arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];

console.log(binarySearch(arr, 5, 0, arr.length - 1)); // 4

Bu yerda:

* **Baza holati**: Agar left > right, qidirish to'xtaydi va -1 qaytariladi.
* **Rekursiv holatlar**: Target qiymati o'rtadagi qiymatdan kichik yoki katta ekanligini tekshirib, mos ravishda yangi oraliqda qidirishni davom ettiradi.

### Rekursiyaning Muhim Asoslari

1. **Baza Holat**: Har qanday rekursiv funksiyada baza holatining bo'lishi kerak. Agar baza holat bo'lmasa, rekursiya cheksiz davom etadi va dastur to'xtaydi (stack overflow).
2. **Progressiya**: Har bir rekursiv chaqiriqda muammo kichikroq bo'lishi kerak. Bu holatda baza holatga erishiladi.
3. **Har bir rekursiv chaqiriqda yangi holat yaratiladi**: Har bir rekursiv chaqiriqda funksiya yangi kontekstda ishlaydi va o'zgaruvchilar o'z holatiga ega bo'ladi.

Rekursiya juda qulay bo'lsa-da, rekursiv funksiyalarni to'g'ri tuzish kerak, chunki noto'g'ri ishlangan rekursiya funktsiyalari katta muammolarni keltirib chiqarishi mumkin.

function fibonacci(n) {

// Baza holatlari

if (n === 0) return 0;

if (n === 1) return 1;

// Rekursiv holat: fibonacci(n - 1) + fibonacci(n - 2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5function fibonacci(n) {

// Baza holatlari

if (n === 0) return 0;

if (n === 1) return 1;

// Rekursiv holat: fibonacci(n - 1) + fibonacci(n - 2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5function fibonacci(n) {

// Baza holatlari

if (n === 0) return 0;

if (n === 1) return 1;

// Rekursiv holat: fibonacci(n - 1) + fibonacci(n - 2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5function fibonacci(n) {

// Baza holatlari

if (n === 0) return 0;

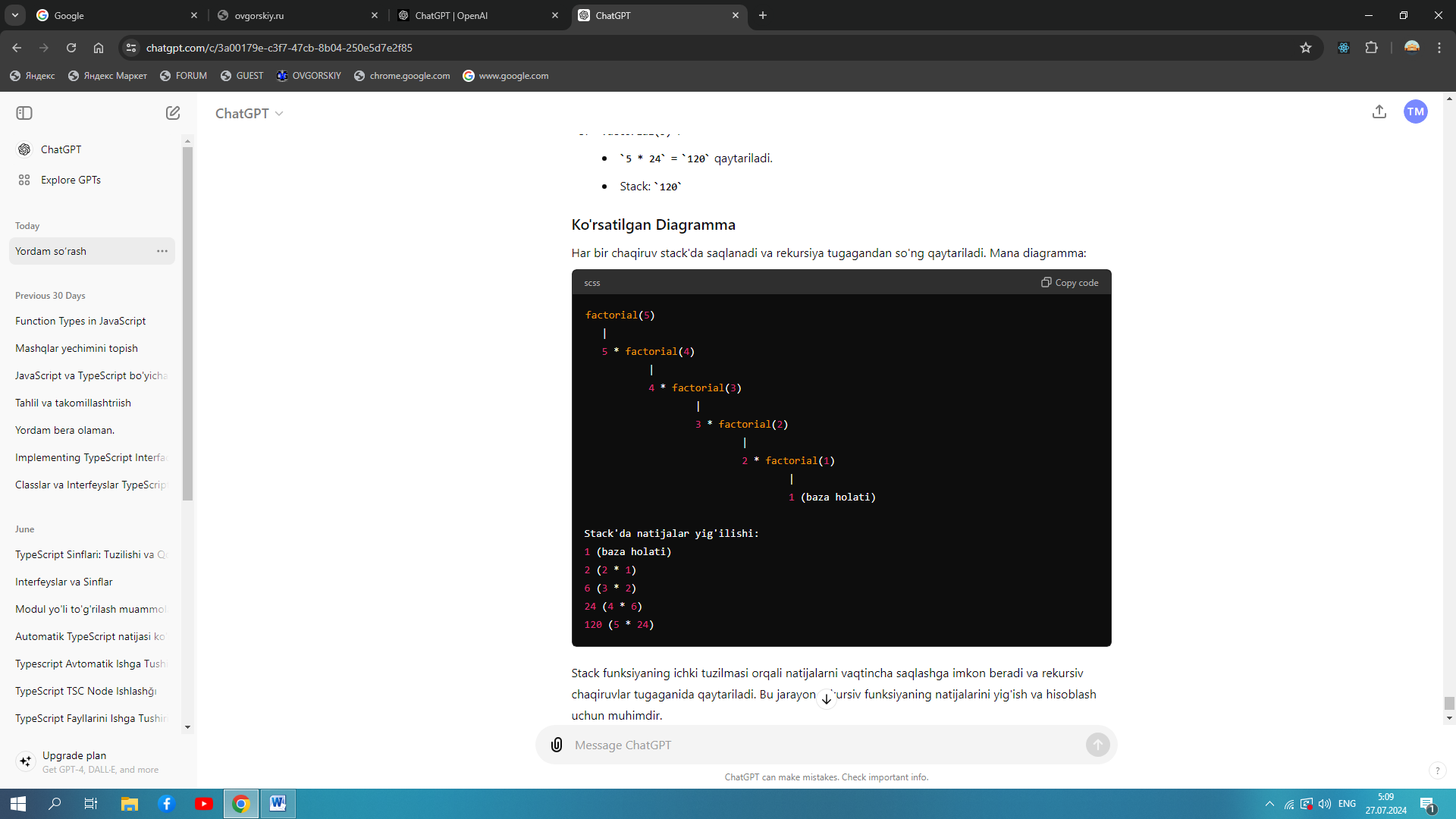
if (n === 1) return 1;

// Rekursiv holat: fibonacci(n - 1) + fibonacci(n - 2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

console.log(fibonacci(5)); // 5

Конец формы