Hands – On Lab Workshop 3.

AREA OF TRIANGLE

Write a function that takes the base and height of a triangle and return its area.

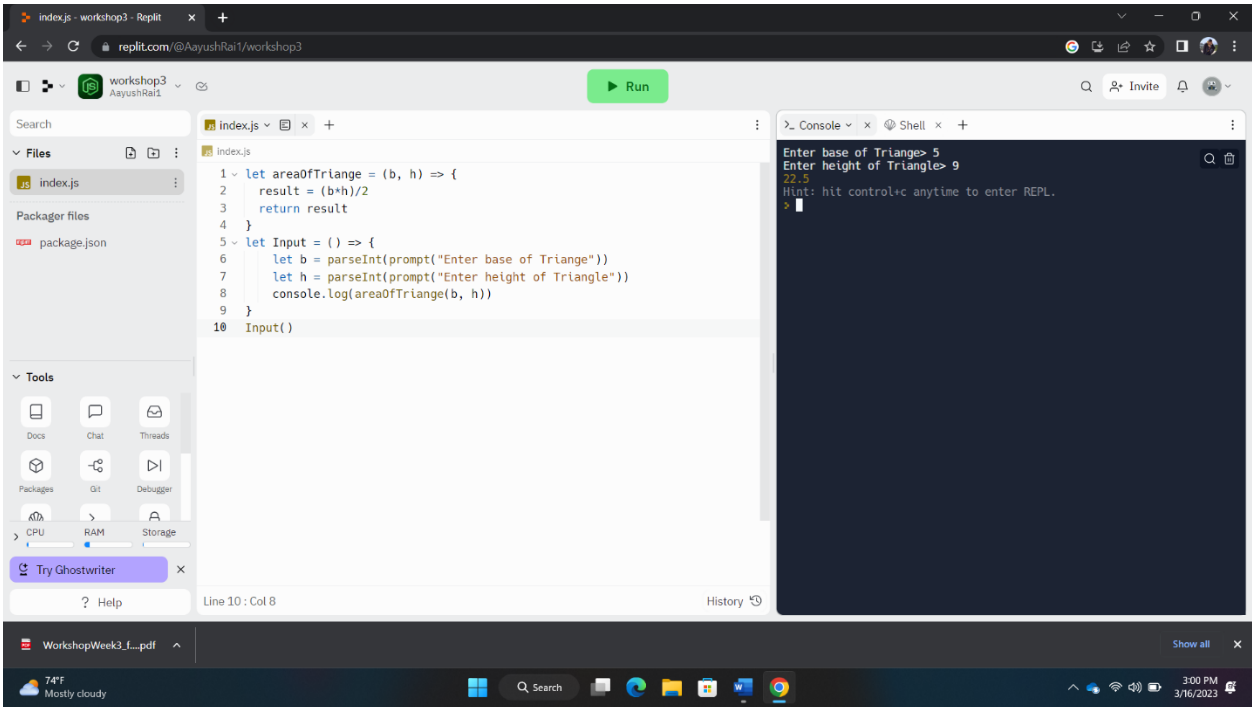
Example:

Areaoftriangle (3, ⟶ 4) 6

Areaoftriangle (7, ⟶ 8) 28

Notes

- Area of triangle is (base * height)/2
- Don’t forget to return the result



```
let areaOfTriange = (b, h) => {
  result = (b*h)/2
  return result
}
let Input = () => {
    let b = parseInt(prompt("Enter base of Triange"))
    let h = parseInt(prompt("Enter height of Triangle"))
    console.log(areaOfTriange(b, h))
}
Input()
```

```
Enter base of Triange> 5
Enter height of Triangle> 9
22.5
Hint: hit control+c anytime to enter REPL.
>
```

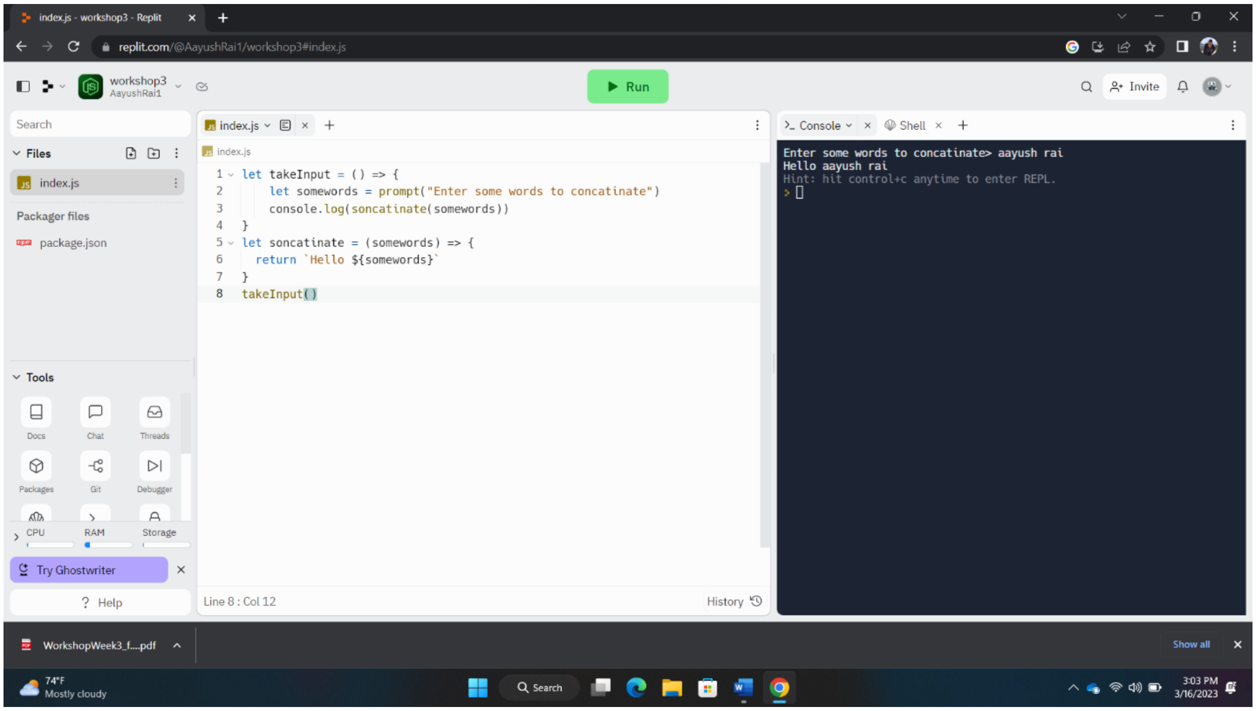
RETURN SOMETHING TO ME!

Write a function that returns the string "something" joined with a space " " and the given argument.

Examples

giveMeSomething("is better than nothing") ➞ "something is better than nothing"

giveMeSomething("Bob Jane") ➞ "something Bob Jane" giveMeSomething("something") ➞ "something something”



```javascript
let takeInput = () => {
    let somewords = prompt("Enter some words to concatinate")
    console.log(soncatinate(somewords))
}
let soncatinate = (somewords) => {
  return `Hello ${somewords}`
}
takeInput()
```

```
Enter some words to concatinate> aayush rai
Hello aayush rai
Hint: hit control+c anytime to enter REPL.
> 
```

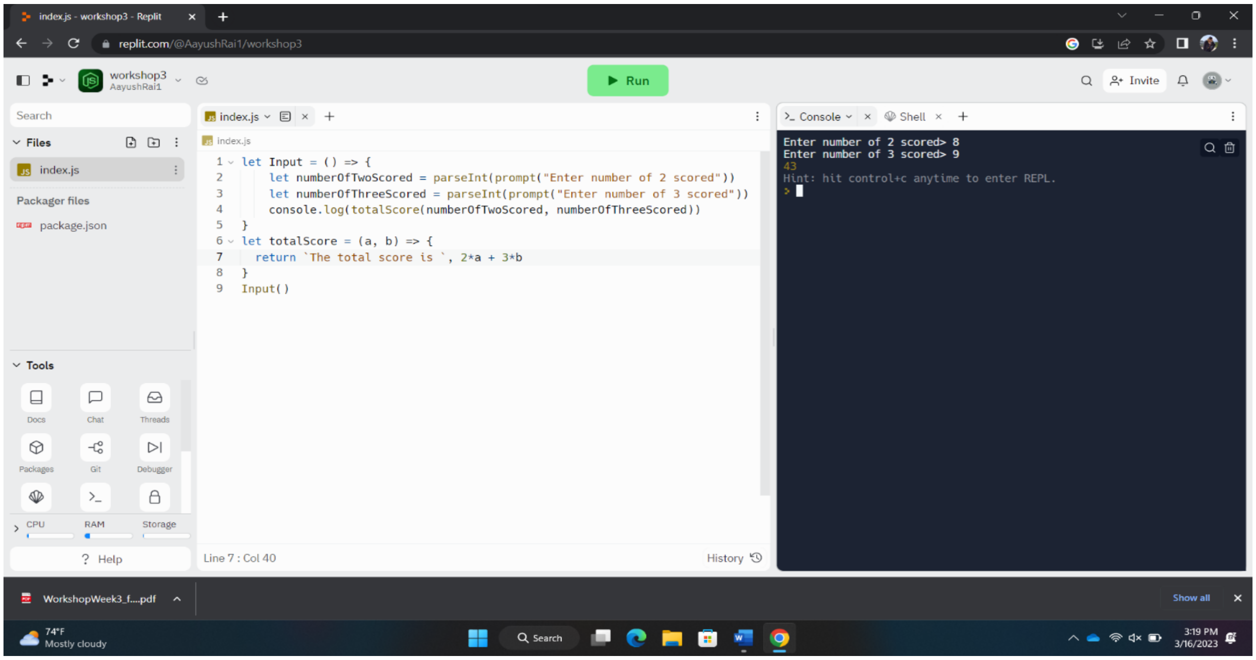
BASKETBALL POINTS

You are counting points for a basketball game, given the amount of 2 – pointer scored and 3 – pointer scored, find the final points for the team and return the value.

Example:

points ⟶ (3,5) 3*2 + 5*3 = 21

points ⟶ (1,1) 5



```javascript
let Input = () => {
    let numberOfTwoScored = parseInt(prompt("Enter number of 2 scored"))
    let numberOfThreeScored = parseInt(prompt("Enter number of 3 scored"))
    console.log(totalScore(numberOfTwoScored, numberOfThreeScored))
}
let totalScore = (a, b) => {
  return `The total score is `, 2*a + 3*b
}
Input()
```

```
Enter number of 2 scored> 8
Enter number of 3 scored> 9
43
Hint: hit control+c anytime to enter REPL.
> 
```

LESS THAN 100?

Given two numbers, return true if the sum of both numbers is less than 100.

Otherwise return false.

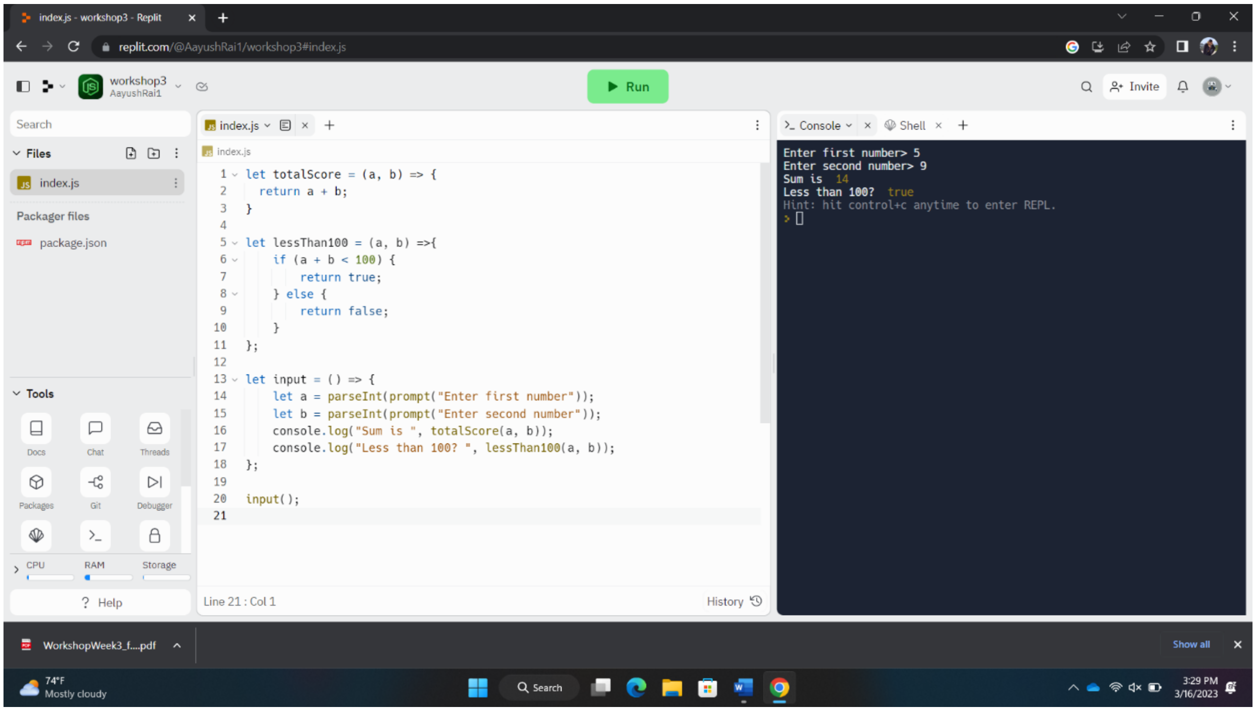
Examples

lessThan100(22, 15) ➞ true

// 22 + 15 = 37 lessThan100(83,

34) ➞ false // 83 + 34 = 117

lessThan100(3, 77) ➞ true

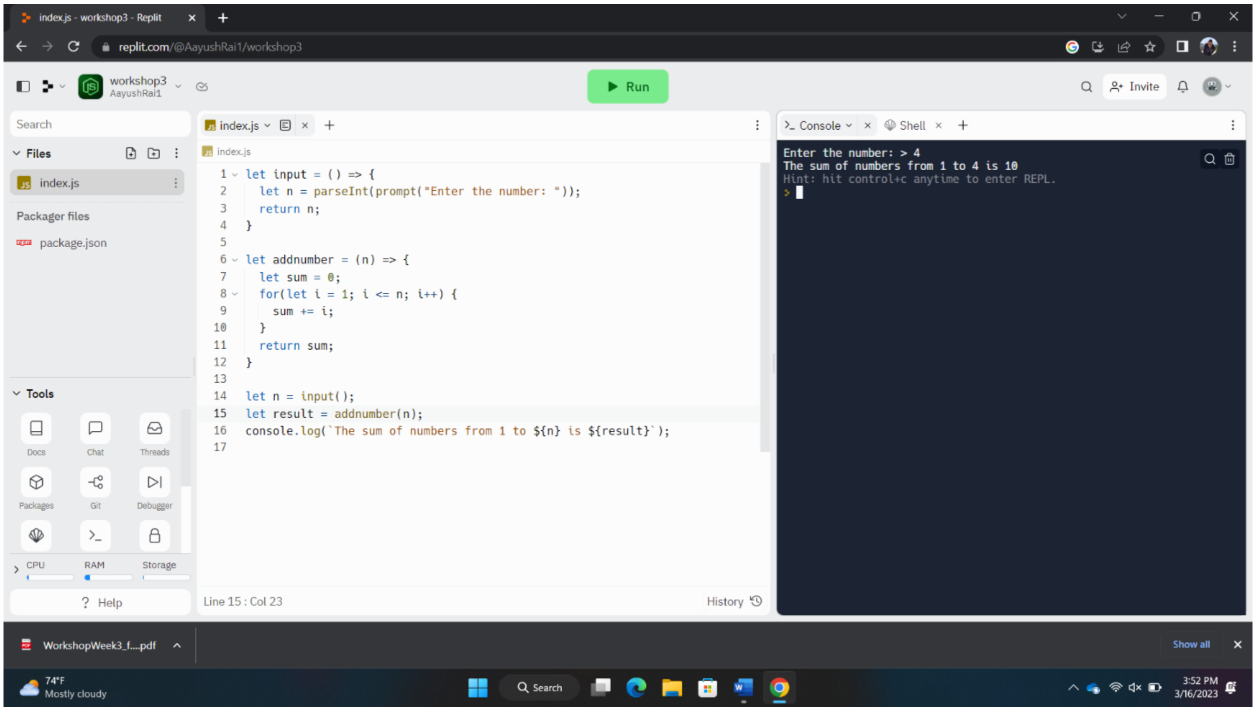


```
let totalScore = (a, b) => {
  return a + b;
}

let lessThan100 = (a, b) =>{
    if (a + b < 100) {
        return true;
    } else {
        return false;
    }
};

let input = () => {
    let a = parseInt(prompt("Enter first number"));
    let b = parseInt(prompt("Enter second number"));
    console.log("Sum is ", totalScore(a, b));
    console.log("Less than 100? ", lessThan100(a, b));
};

input();
```

```
Enter first number> 5
Enter second number> 9
Sum is  14
Less than 100?  true
Hint: hit control+c anytime to enter REPL.
> 
```

ADD UPTO THE NUMBER FROM A SINGLE NUMBER

Create a function that takes a number as an argument. Add up all the numbers from 1 to the number you passed to the function. For example, if the input is 4 then your function should return 10 because 1+2+3+4 = 10



```javascript
let input = () => {
  let n = parseInt(prompt("Enter the number: "));
  return n;
}

let addnumber = (n) => {
  let sum = 0;
  for(let i = 1; i <= n; i++) {
    sum += i;
  }
  return sum;
}

let n = input();
let result = addnumber(n);
console.log(`The sum of numbers from 1 to ${n} is ${result}`);
```

```
Enter the number: > 4
The sum of numbers from 1 to 4 is 10
Hint: hit control+c anytime to enter REPL.
>
```

ANY PRIME NUMBER IN RANGE

Create a function that return true if there is at least one prime number in the given range(n1 to n2) inclusive, false otherwise.

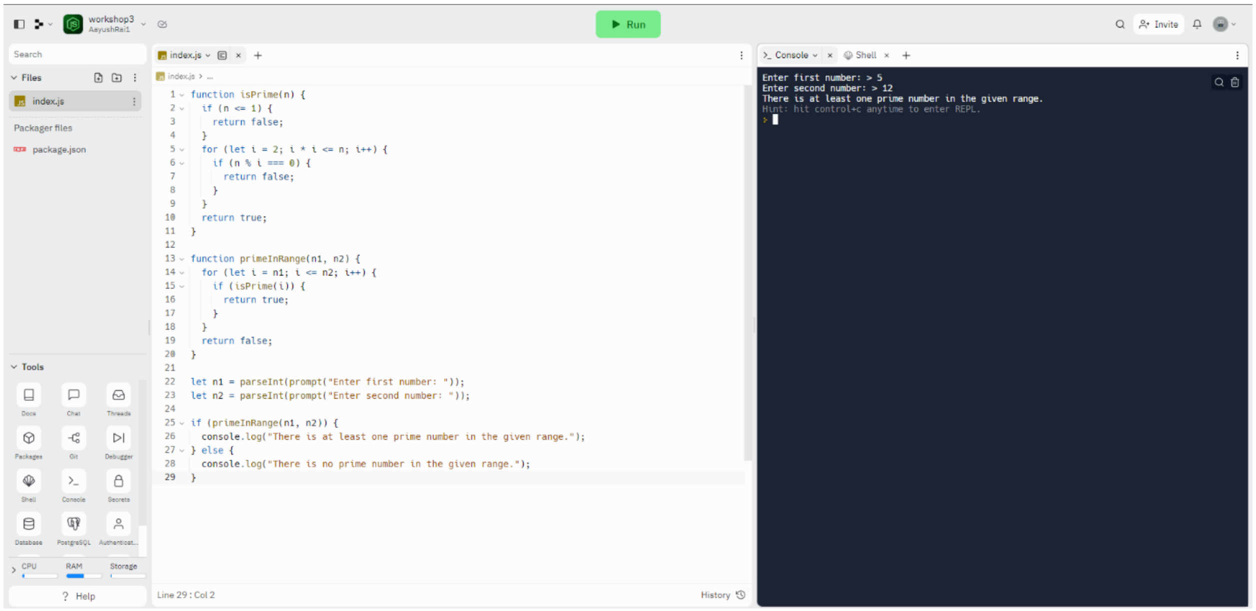
Example: primeInRange(10,15)

⟶ true

// prime number is range : 11, 13 primeInRange(3,1)

⟶ true

// prime number is range : 3, 5



```javascript
function isPrime(n) {
  if (n <= 1) {
    return false;
  }
  for (let i = 2; i * i <= n; i++) {
    if (n % i === 0) {
      return false;
    }
  }
  return true;
}

function primeInRange(n1, n2) {
  for (let i = n1; i <= n2; i++) {
    if (isPrime(i)) {
      return true;
    }
  }
  return false;
}

let n1 = parseInt(prompt("Enter first number: "));
let n2 = parseInt(prompt("Enter second number: "));

if (primeInRange(n1, n2)) {
  console.log("There is at least one prime number in the given range.");
} else {
  console.log("There is no prime number in the given range.");
}
```

```
Enter first number: > 5
Enter second number: > 12
There is at least one prime number in the given range.
Hint: hit control+c anytime to enter REPL.
>
```

ODDISH VS. EVENISH

Create a function that determines whether a number is Oddish or Evenish. A number is Oddish if the sum of all of its digits is odd, and a number is Evenish if the sum of all of its digits is even. If a number is Oddish, return "Oddish". Otherwise, return "Evenish".

For example, oddishOrEvenish(121) should return "Evenish", since 1 + 2 + 1 = 4. oddishOrEvenish(41) should return "Oddish", since 4 + 1 = 5.

Examples

oddishOrEvenish(43) ➞ "Oddish"

// 4 + 3 = 7 //

7 % 2 = 1

oddishOrEvenish(373) ➞ "Oddish"

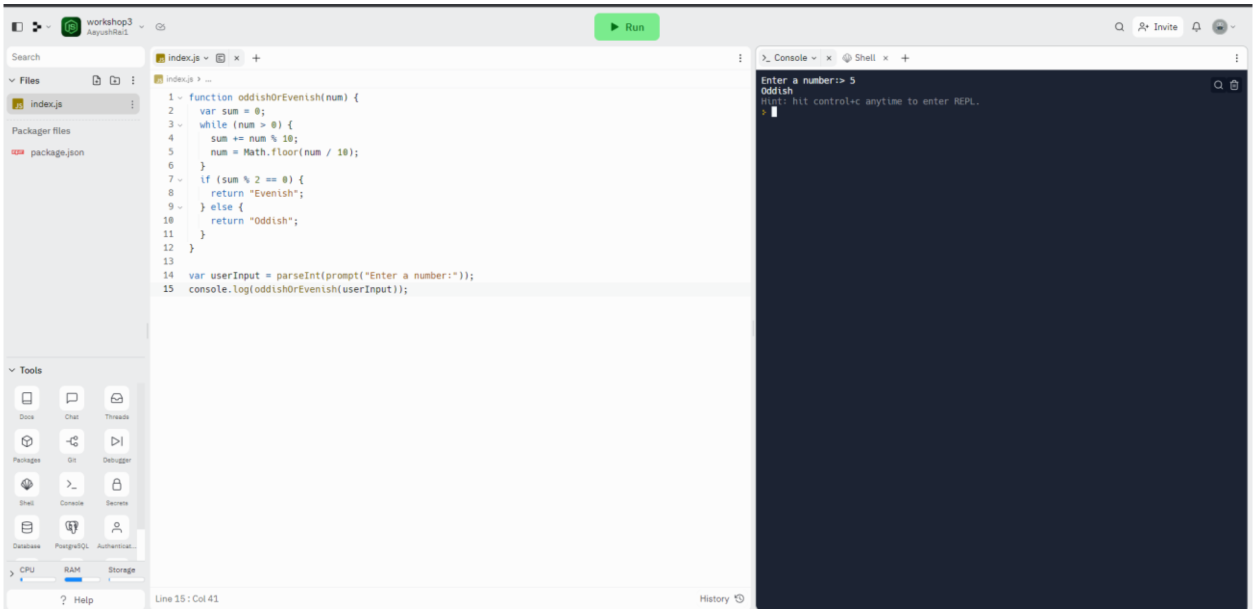
// 3 + 7 + 3 = 13 //

13 % 2 = 1

oddishOrEvenish(4433) ➞ "Evenish"

// 4 + 4 + 3 + 3 = 14

// 14 % 2 = 0



```
function oddishOrEvenish(num) {
  var sum = 0;
  while (num > 0) {
    sum += num % 10;
    num = Math.floor(num / 10);
  }
  if (sum % 2 == 0) {
    return "Evenish";
  } else {
    return "Oddish";
  }
}

var userInput = parseInt(prompt("Enter a number:"));
console.log(oddishOrEvenish(userInput));
```

```
Enter a number:> 5
Oddish
Hint: hit control+c anytime to enter REPL.
> 
```

LEFT SHIFT BY POWERS OF TWO

The left shift operation is similar to multiplication by powers oftwo.

Sample calculation using the left shift operator (<<):

10 << 3 = 10 * 2^3 = 10 * 8 = 80

-32 << 2 = -32 * 2^2 = -32 * 4 = -128

5 << 2 = 5 * 2^2 = 5 * 4 = 20

Write a function that mimics (withoutthe use of <<)the left shift operator and returns the resultfrom the two given integers.

Examples

shiftToLeft(5, 2) ➞ 20

shiftToLeft(10, 3) ➞ 80

shiftToLeft(-32, 2) ➞ -128

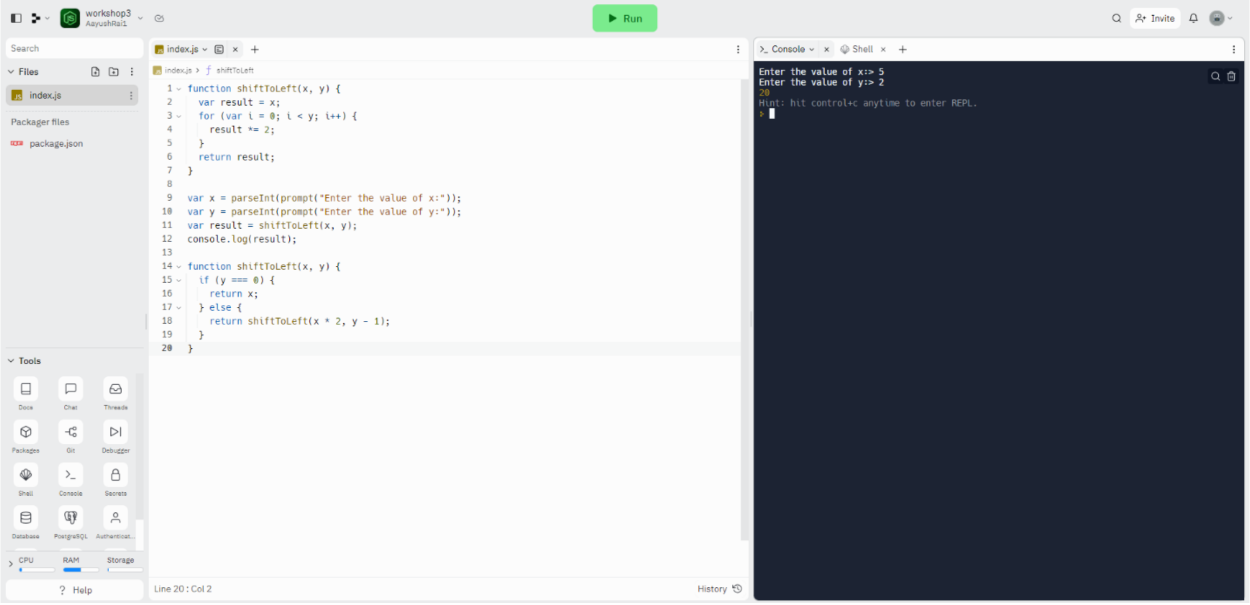
shiftToLeft(-6, 5) ➞ -192

shiftToLeft(12, 4) ➞ 192

shiftToLeft(46, 6) ➞ 2944 Notes

● There will be no negative values for the second parameter y.

- This challenge is more like recreating the left shift operation,thus,the use ofthe operator directly is prohibited.
- Alternatively, you can solve this challenge via recursion.



```javascript
function shiftToLeft(x, y) {
  var result = x;
  for (var i = 0; i < y; i++) {
    result *= 2;
  }
  return result;
}

var x = parseInt(prompt("Enter the value of x:"));
var y = parseInt(prompt("Enter the value of y:"));
var result = shiftToLeft(x, y);
console.log(result);

function shiftToLeft(x, y) {
  if (y === 0) {
    return x;
  } else {
    return shiftToLeft(x * 2, y - 1);
  }
}
```

```
Enter the value of x:> 5
Enter the value of y:> 2
20
Hint: hit control+c anytime to enter REPL.
> 
```

CONVERT A NUMBER TO BASE-2

Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convertis simple: ((2) means base-2 and (10) means base-10) 010101001(2) = 1 + 8 + 32 + 128.

Going from rightto left,the value ofthe most right bitis 1, now from that every bitto the left will be x2. The values of an 8 bit binary number are (256, 128, 64, 32, 16, 8, 4, 2, 1).

Examples

binary(1) ➞ "1"

// 1*1 = 1 binary(5)

➞ "101" // 1*1 +

1*4 = 5 binary(10)

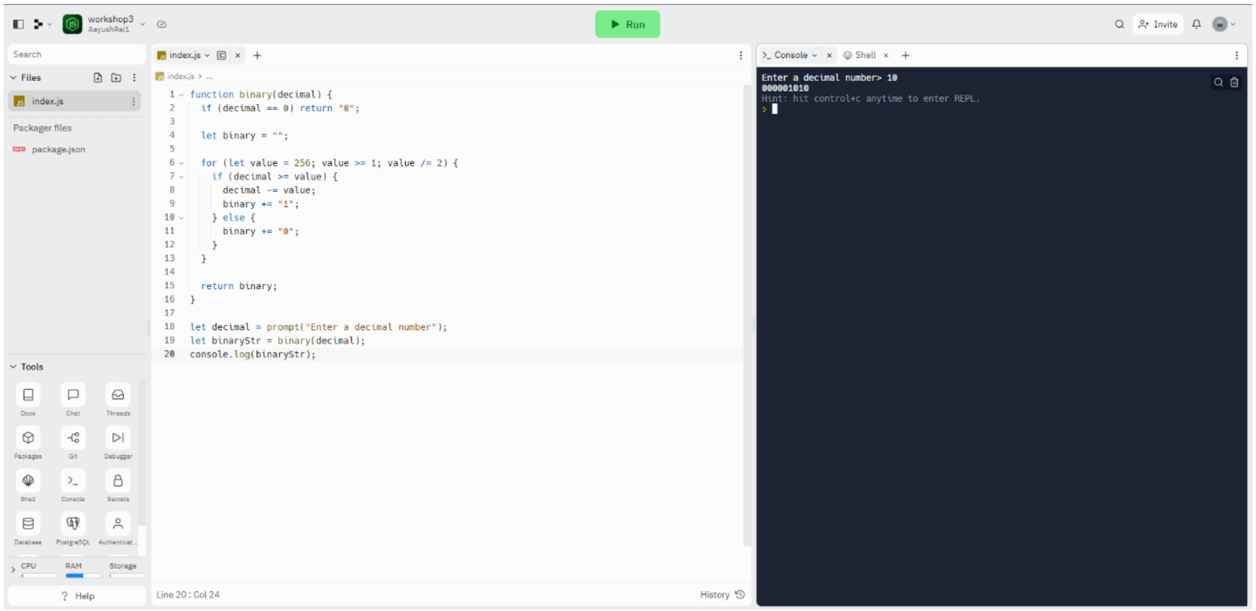
➞ "1010"

// 1*2 + 1*8 = 10

Notes

- Numbers will always be below 1024 (notincluding 1024).

- The && operator could be useful.
- The strings will always go to the length at which the mostleft bit's value gets bigger than the number in decimal.
- If a binary conversion for 0 is attempted, return "0".



```javascript
function binary(decimal) {
  if (decimal == 0) return "0";

  let binary = "";

  for (let value = 256; value >= 1; value /= 2) {
    if (decimal >= value) {
      decimal -= value;
      binary += "1";
    } else {
      binary += "0";
    }
  }

  return binary;
}

let decimal = prompt("Enter a decimal number");
let binaryStr = binary(decimal);
console.log(binaryStr);
```

```
Enter a decimal number> 10
000001010
Hint: hit control+c anytime to enter REPL.
> 
```

GUESSING GAME

Generate a random number (do research) and store it in a variable. Write a program to take input from the user and tell them whether their guessed number is correct, greater or lesser than the original number. (100 – number of guesses) is the score of user. The program is expected to terminate once the number is guessed. Number should be between 1 – 100.

Example:

Random number generated by computer: 54

User input: 34

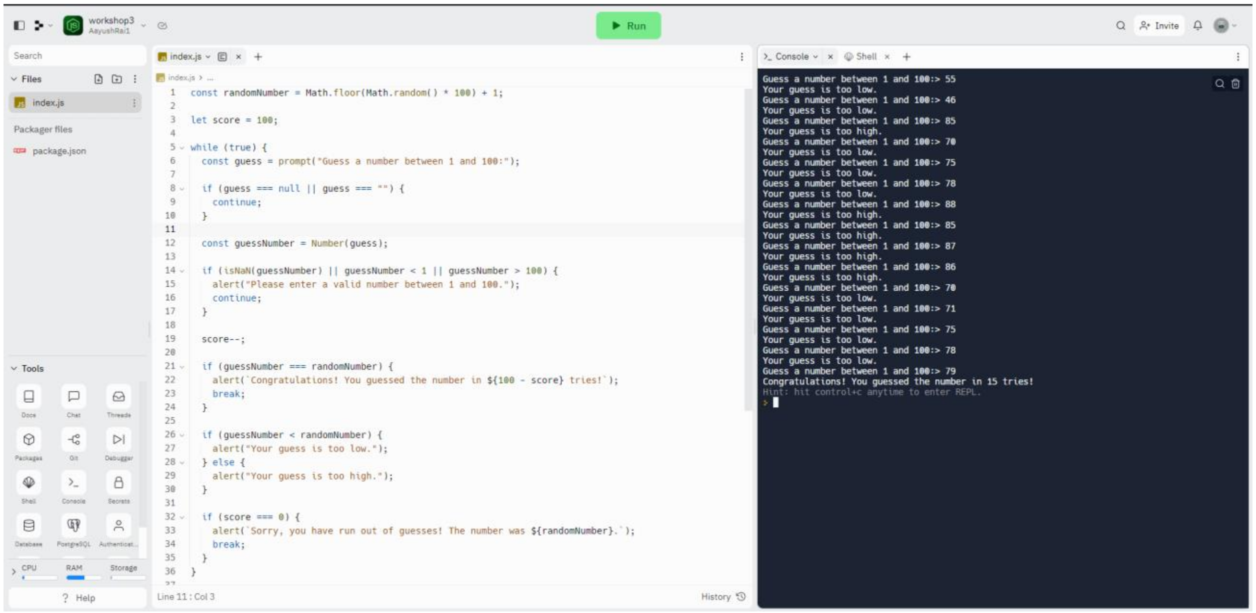
// lesser than original number

User input: 67

// greater than original number

User input: 54

// congratulations!!! The number you guessed matched the original number. Your score is 97!



```javascript
const randomNumber = Math.floor(Math.random() * 100) + 1;

let score = 100;

while (true) {
  const guess = prompt("Guess a number between 1 and 100:");

  if (guess === null || guess === "") {
    continue;
  }

  const guessNumber = Number(guess);

  if (isNaN(guessNumber) || guessNumber < 1 || guessNumber > 100) {
    alert("Please enter a valid number between 1 and 100.");
    continue;
  }

  score--;

  if (guessNumber === randomNumber) {
    alert(`Congratulations! You guessed the number in ${100 - score} tries!`);
    break;
  }

  if (guessNumber < randomNumber) {
    alert("Your guess is too low.");
  } else {
    alert("Your guess is too high.");
  }

  if (score === 0) {
    alert(`Sorry, you have run out of guesses! The number was ${randomNumber}.`);
    break;
  }
}
```

```
Guess a number between 1 and 100:> 55
Your guess is too low.
Guess a number between 1 and 100:> 46
Your guess is too low.
Guess a number between 1 and 100:> 85
Your guess is too high.
Guess a number between 1 and 100:> 70
Your guess is too low.
Guess a number between 1 and 100:> 75
Your guess is too low.
Guess a number between 1 and 100:> 78
Your guess is too low.
Guess a number between 1 and 100:> 88
Your guess is too high.
Guess a number between 1 and 100:> 85
Your guess is too high.
Guess a number between 1 and 100:> 87
Your guess is too high.
Guess a number between 1 and 100:> 86
Your guess is too high.
Guess a number between 1 and 100:> 70
Your guess is too low.
Guess a number between 1 and 100:> 71
Your guess is too low.
Guess a number between 1 and 100:> 75
Your guess is too low.
Guess a number between 1 and 100:> 78
Your guess is too low.
Guess a number between 1 and 100:> 79
Congratulations! You guessed the number in 15 tries!
Hint: hit control+c anytime to enter REPL.
>
```

HIGHER ORDER ARRAY METHODS

Const age = [23,34,12,54,23,54,11,9,29,17,15,19,20,21,13,7]

a. Filter the array of age who can apply for citizenships
b. Find the average age of a given array Const companies = [

{ name: "ABC", category: "Finance", start: 1981, end: 2004 },

{ name: "XYZ", category: "Retail", start: 1991, end: 20012 },

{ name: "DGF", category: "Finance", start: 1976, end: 2008 },

{ name: "LFT", category: "Retail", start: 1971, end: 1979 },

{ name: "MND", category: "Retail", start: 1995, end: 2010 },

{ name: "HCK", category: "Technology", start: 1987, end: 2011 },

{ name: "BMC", category: "Technology", start: 1989, end: 2009 },

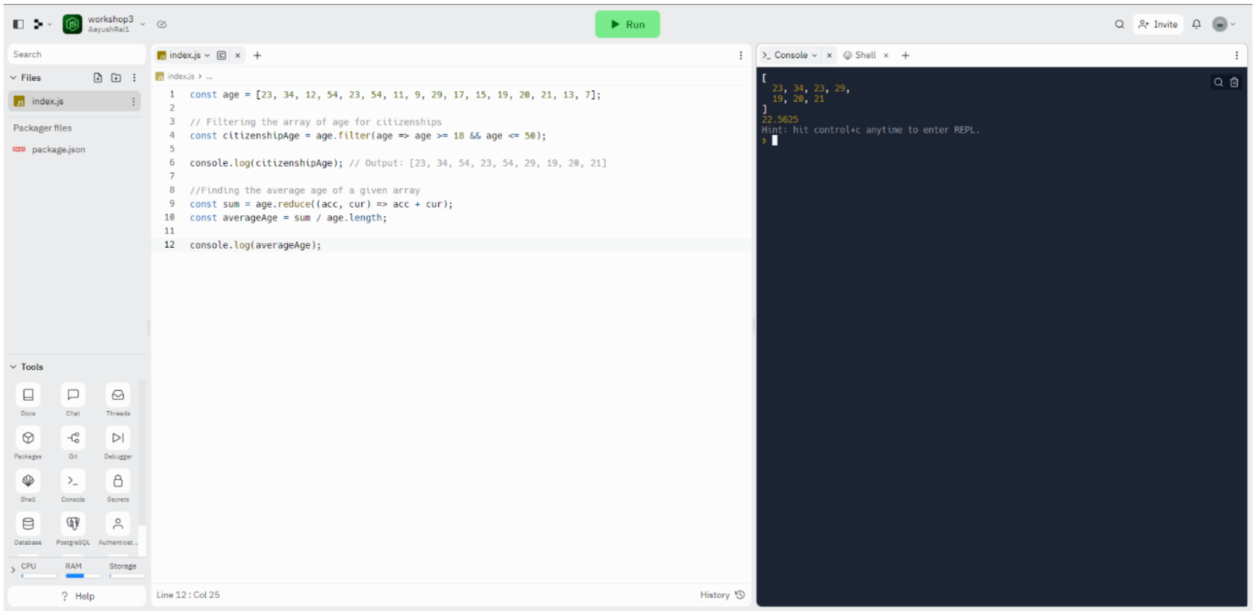
{ name: "TIC", category: "Retail", start: 1993, end: 2005 },

{ name: "NAC", category: "Technology", start: 1991, end: 2010 },

{ name: "ITC", category: "Finance", start: 1998, end: 2016 }

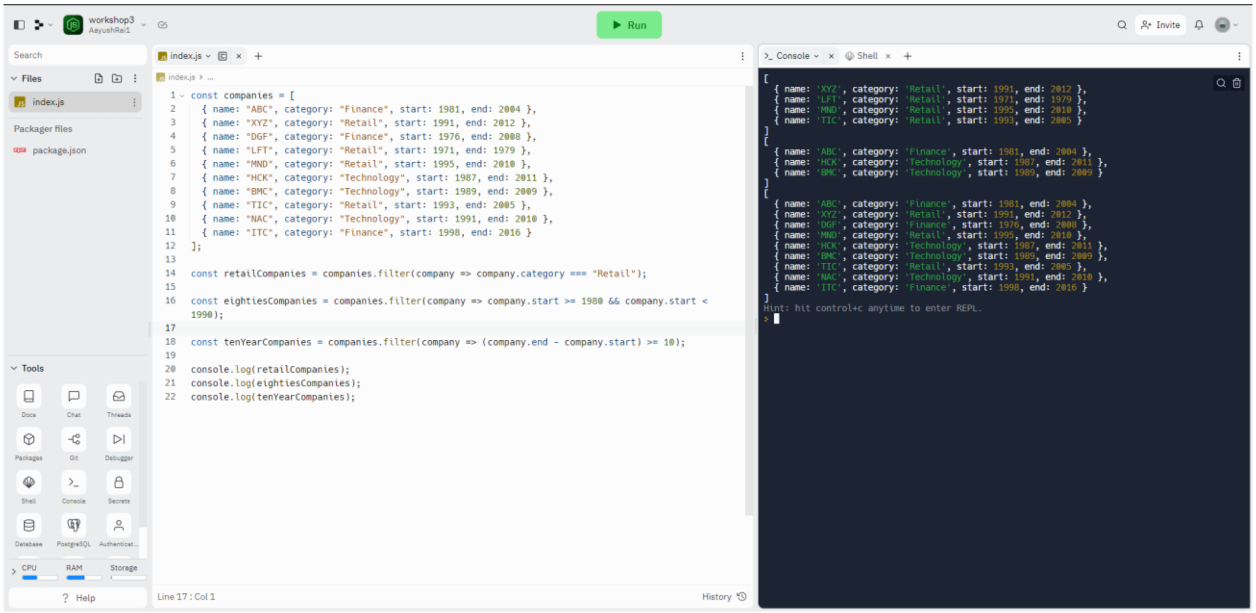
];

a. Filter the retail companies
b. Get the 80s companies from the array
c. Get the companies that lasted for 10 or more years



```javascript
const age = [23, 34, 12, 54, 23, 54, 11, 9, 29, 17, 15, 19, 20, 21, 13, 7];

// Filtering the array of age for citizenships
const citizenshipAge = age.filter(age => age >= 18 && age <= 50);

console.log(citizenshipAge); // Output: [23, 34, 54, 23, 54, 29, 19, 20, 21]

//Finding the average age of a given array
const sum = age.reduce((acc, cur) => acc + cur);
const averageAge = sum / age.length;

console.log(averageAge);
```

```
[
  23, 34, 23, 29,
  19, 20, 21
]
22.5625
Hint: hit control+c anytime to enter REPL.
> 
```



```javascript
const companies = [
  { name: "ABC", category: "Finance", start: 1981, end: 2004 },
  { name: "XYZ", category: "Retail", start: 1991, end: 2012 },
  { name: "DGF", category: "Finance", start: 1976, end: 2008 },
  { name: "LFT", category: "Retail", start: 1971, end: 1979 },
  { name: "MND", category: "Retail", start: 1995, end: 2010 },
  { name: "HCK", category: "Technology", start: 1987, end: 2011 },
  { name: "BMC", category: "Technology", start: 1989, end: 2009 },
  { name: "TIC", category: "Retail", start: 1993, end: 2005 },
  { name: "NAC", category: "Technology", start: 1991, end: 2010 },
  { name: "ITC", category: "Finance", start: 1998, end: 2016 }
];

const retailCompanies = companies.filter(company => company.category === "Retail");

const eightiesCompanies = companies.filter(company => company.start >= 1980 && company.start < 1990);

const tenYearCompanies = companies.filter(company => (company.end - company.start) >= 10);

console.log(retailCompanies);
console.log(eightiesCompanies);
console.log(tenYearCompanies);
```

```
[
  { name: 'XYZ', category: 'Retail', start: 1991, end: 2012 },
  { name: 'LFT', category: 'Retail', start: 1971, end: 1979 },
  { name: 'MND', category: 'Retail', start: 1995, end: 2010 },
  { name: 'TIC', category: 'Retail', start: 1993, end: 2005 }
]
[
  { name: 'ABC', category: 'Finance', start: 1981, end: 2004 },
  { name: 'HCK', category: 'Technology', start: 1987, end: 2011 },
  { name: 'BMC', category: 'Technology', start: 1989, end: 2009 }
]
[
  { name: 'ABC', category: 'Finance', start: 1981, end: 2004 },
  { name: 'XYZ', category: 'Retail', start: 1991, end: 2012 },
  { name: 'DGF', category: 'Finance', start: 1976, end: 2008 },
  { name: 'MND', category: 'Retail', start: 1995, end: 2010 },
  { name: 'HCK', category: 'Technology', start: 1987, end: 2011 },
  { name: 'BMC', category: 'Technology', start: 1989, end: 2009 },
  { name: 'TIC', category: 'Retail', start: 1993, end: 2005 },
  { name: 'NAC', category: 'Technology', start: 1991, end: 2010 },
  { name: 'ITC', category: 'Finance', start: 1998, end: 2016 }
]
Hint: hit control+c anytime to enter REPL.
> 
```