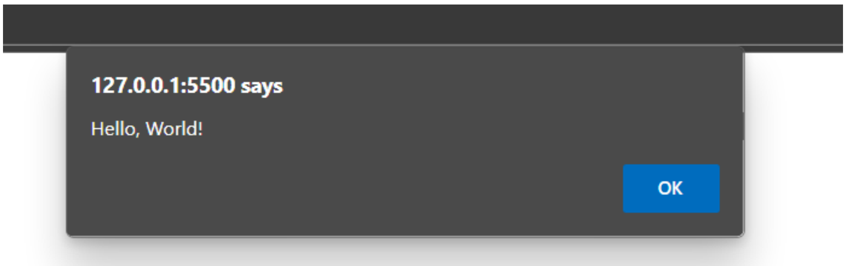
717822F123

TASK 1:

Code:

```
 <script>
       alert("Hello, World!")
   </script>
```

Output:



127.0.0.1:5500 says

Hello, World!

TASK 2:

Code:

```
let Name="KAVI";
let RollNo=123;
let Pass=true;
let Age=null;
let Dob;
let Bigger=98765432109876543210987654321098765432 1n;
let user={
    name:"Kavi",
    age:20
};
let mySymbol = Symbol('id');
let obj = {
  [mySymbol]: 'valueForSymbol'
};
console.log(Name);
console.log(RollNo);
```

console.log(Pass);

console.log(Age);

console.log(Dob);

console.log(Bigger);

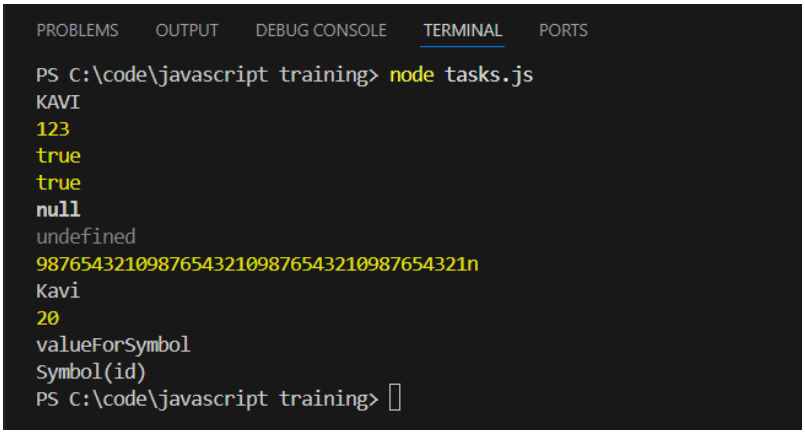
console.log(user.name);

console.log(user.age);

console.log(obj[mySymbol]);

console.log(mySymbol.toString());

Output:



```
PS C:\code\javascript training> node tasks.js
KAVI
123
true
true
null
undefined
987654321098765432109876543210987654321n
Kavi
20
valueForSymbol
Symbol(id)
PS C:\code\javascript training> 
```

TASK 3:

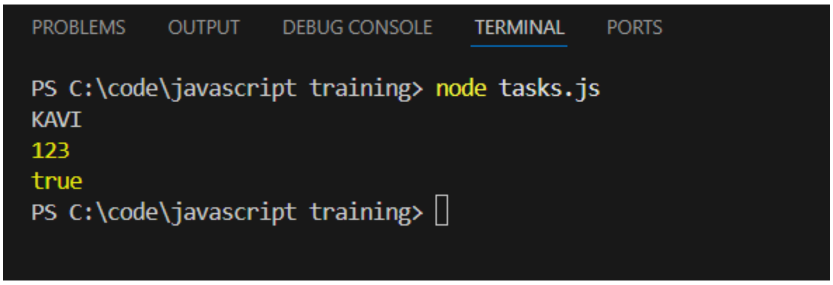
Code:

let str1="Kavi";

let str2="tha";

console.log(str1+str2);

Output:



```
PS C:\code\javascript training> node tasks.js
KAVI
123
true
PS C:\code\javascript training> 
```

TASK 4:

Code:

let a=10;

let b=5;

let c=0;

let d=2;

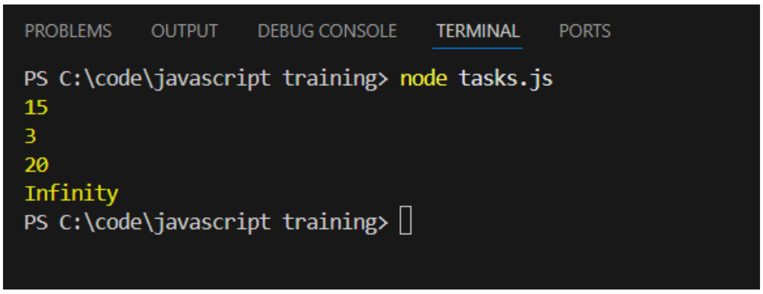
console.log(a+b);

console.log(b-d);

console.log(a*d);

console.log(b/c);

Output:



```
PS C:\code\javascript training> node tasks.js
15
3
20
Infinity
PS C:\code\javascript training> 
```

TASK 5:

Code:

let Name="KAVI";

let RollNo=123;

let Pass=true;

let Age=null;

let Dob;

let Bigger=987654321098765432109876543210987654321n;

let user={

  name:"Kavi",

  age:20

};

let mySymbol = Symbol('id');

let obj = {

 [mySymbol]: 'valueForSymbol'

};

console.log(typeof(Name));

console.log(typeof(RollNo));

console.log(typeof(Pass));

console.log(typeof(Age));

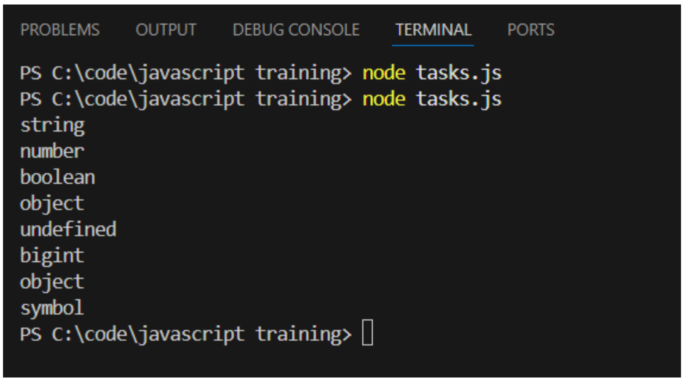
console.log(typeof(Dob));

console.log(typeof(Bigger));

console.log(typeof(user));

console.log(typeof(mySymbol));

Output:



```
PS C:\code\javascript training> node tasks.js
PS C:\code\javascript training> node tasks.js
string
number
boolean
object
undefined
bigint
object
symbol
PS C:\code\javascript training> 
```

TASK 6:

Code:

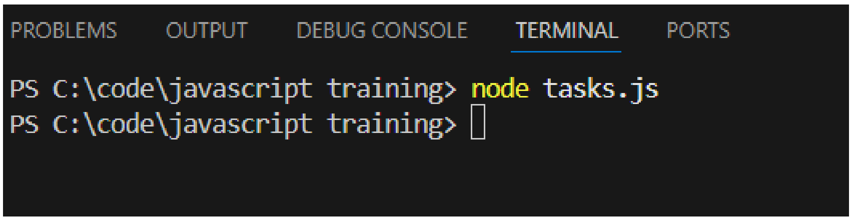
/*

Multiline comments are comments that has more lines taken as an instruction to the compiler and it is skipped to be executed.Anything between (/* and *slash) is treated as a comment.Multi-line comments are used for detailed documentation or temporarily disabling large blocks of code.

*/

//In single line comments everything after // on the same line is treated as a comment and ignored by the JavaScript engine.

Output:



```
PS C:\code\javascript training> node tasks.js
PS C:\code\javascript training> 
```

TASK 7:

In JavaScript, semicolons (;) are used to terminate statements, but the language includes an automatic semicolon insertion (ASI) feature, which allows semicolons to be omitted in many cases.

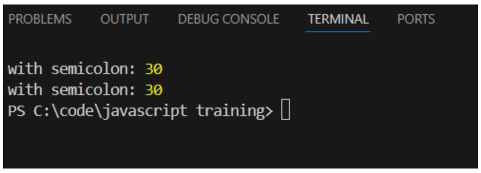
Code:

```
let a =10;

let b=20;

let c=a+b;

console.log("with semicolon:",c);

let x =10

let y=20

let z=x+y

console.log("without semicolon:",z);
```

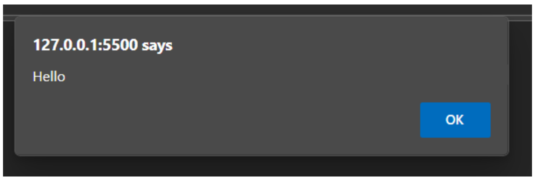
semicolon required case:

```
 alert("Hello");

[1,2].forEach(alert);
```

Output:

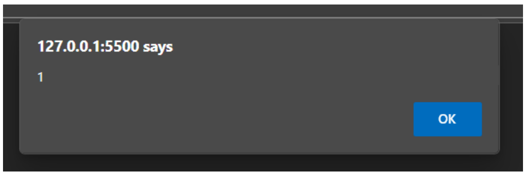


```
with semicolon: 30
with semicolon: 30
PS C:\code\javascript training> 
```



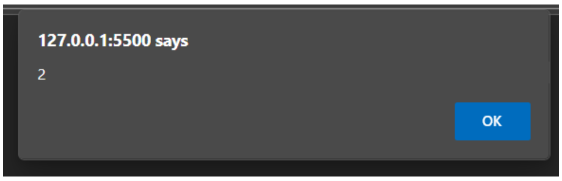
127.0.0.1:5500 says

Hello



127.0.0.1:5500 says

1



127.0.0.1:5500 says

2

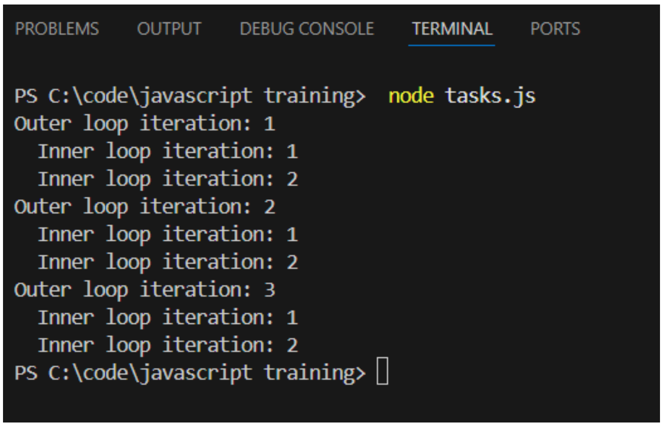
TASK 8:

Code:

```
for (let i = 1; i <= 3; i++) {

   console.log(`Outer loop iteration: ${i}`);

   for (let j = 1; j <= 2; j++) {

      console.log(`  Inner loop iteration: ${j}`);

   }
```

}

Output:



```
PS C:\code\javascript training>  node tasks.js
Outer loop iteration: 1
  Inner loop iteration: 1
  Inner loop iteration: 2
Outer loop iteration: 2
  Inner loop iteration: 1
  Inner loop iteration: 2
Outer loop iteration: 3
  Inner loop iteration: 1
  Inner loop iteration: 2
PS C:\code\javascript training> 
```

TASK 9:

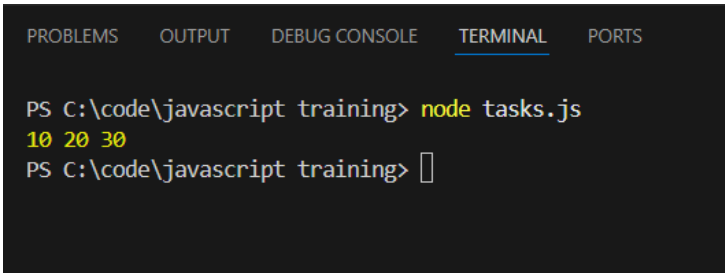
Code:

let x, y, z;

x = 10; y = 20; z = 30;

console.log(x, y, z);

Output:



```
PS C:\code\javascript training> node tasks.js
10 20 30
PS C:\code\javascript training> 
```

TASK 10:

Code:

<script> within head:

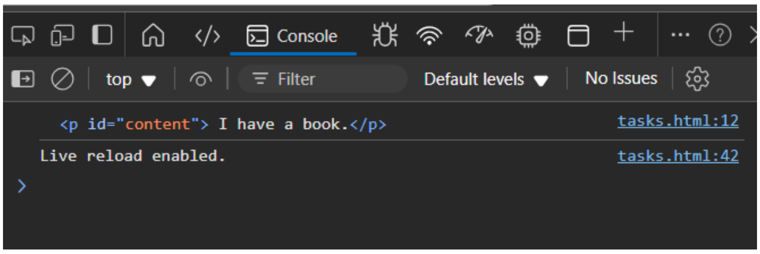
```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    console.log(document.getElementById("content"));
  </script>
```

```
</head>
<body>
  <p id ="content"> I have a book.</p>
</body>
```

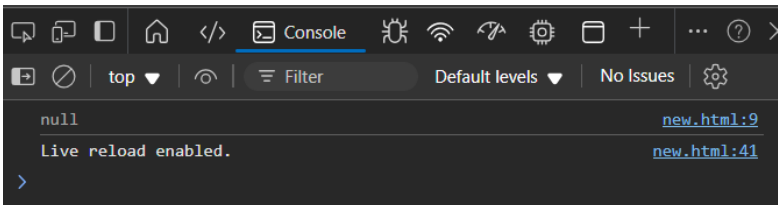
<script> within body:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p id ="content"> I have a book.</p>
  <script>
    console.log(document.getElementById("content"));
  </script>
</body>
```

Output:



```
<p id="content"> I have a book.</p>          tasks.html:12
Live reload enabled.                           tasks.html:42
>
```



```
null                                           new.html:9
Live reload enabled.                           new.html:41
>
```

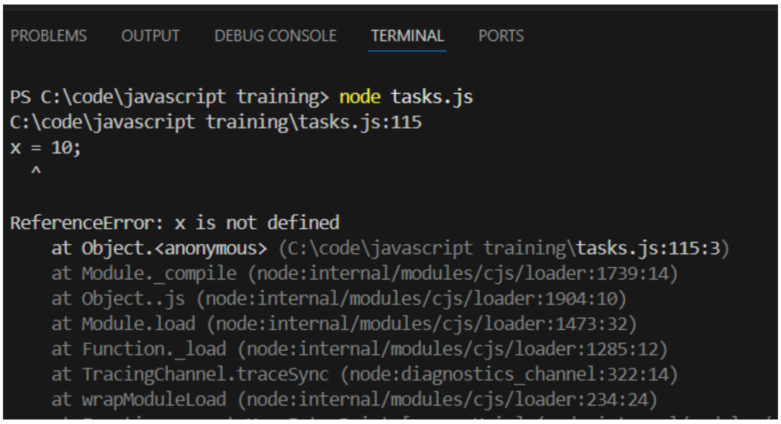
TASK 11:

Code:

```
'use strict';
x = 10;
```

console.log(x);

Output



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:115
x = 10;
  ^

ReferenceError: x is not defined
    at Object.<anonymous> (C:\code\javascript training\tasks.js:115:3)
    at Module._compile (node:internal/modules/cjs/loader:1739:14)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:234:24)
```

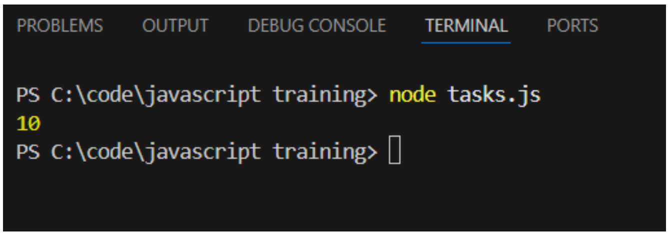
TASK 12:

Code:

x = 10;

console.log(x);

Output:



```
PS C:\code\javascript training> node tasks.js
10
PS C:\code\javascript training> 
```

TASK 13:

Code:

'use strict';

let x = 10;

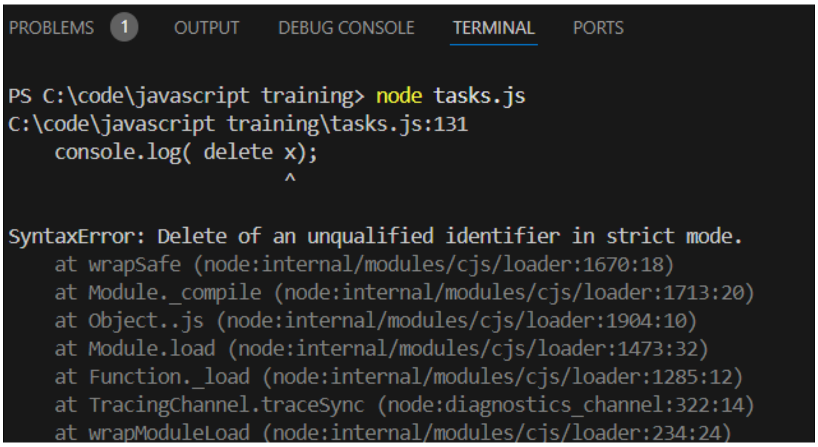
function user(x){

  console.log(delete x);

}

user(20);

Output:



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:131
    console.log( delete x);
                        ^

SyntaxError: Delete of an unqualified identifier in strict mode.
    at wrapSafe (node:internal/modules/cjs/loader:1670:18)
    at Module._compile (node:internal/modules/cjs/loader:1713:20)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:234:24)
```

TASK 14:

Code:

Without using strict:

x = 10;

console.log(x);

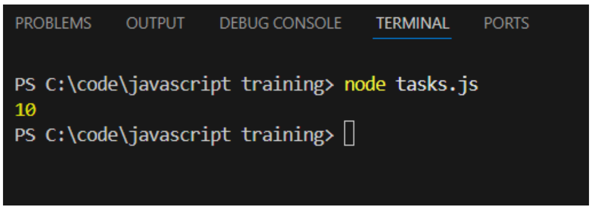
Using strict:

'use strict';

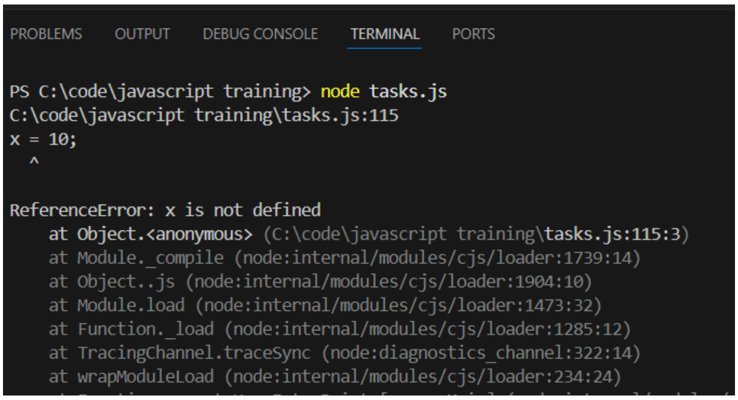
x = 10;

console.log(x);

Output:



```
PS C:\code\javascript training> node tasks.js
10
PS C:\code\javascript training> 
```



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:115
x = 10;
  ^

ReferenceError: x is not defined
    at Object.<anonymous> (C:\code\javascript training\tasks.js:115:3)
    at Module._compile (node:internal/modules/cjs/loader:1739:14)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:234:24)
```

TASK 15:

Code:

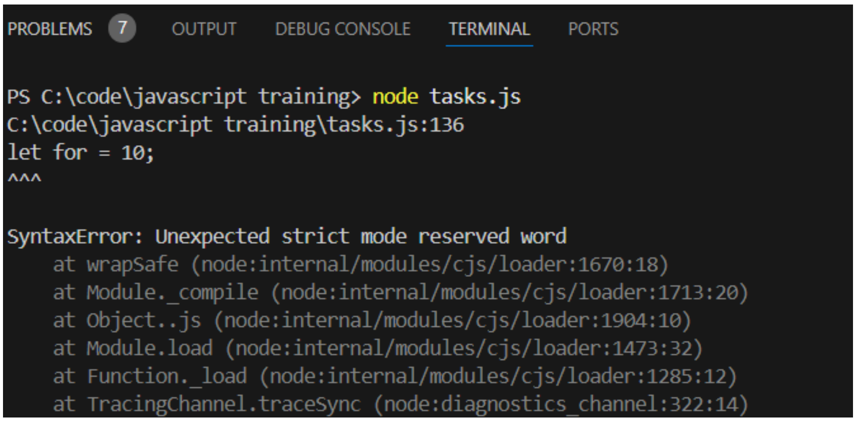
'use strict';

let for = 10;

let while=20;

console.log(for);

Output:



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:136
let for = 10;
^^^

SyntaxError: Unexpected strict mode reserved word
    at wrapSafe (node:internal/modules/cjs/loader:1670:18)
    at Module._compile (node:internal/modules/cjs/loader:1713:20)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
```

TASK 16:

Code:

let a=10;

var b=20;

const c=30;

a=11;

b=21;

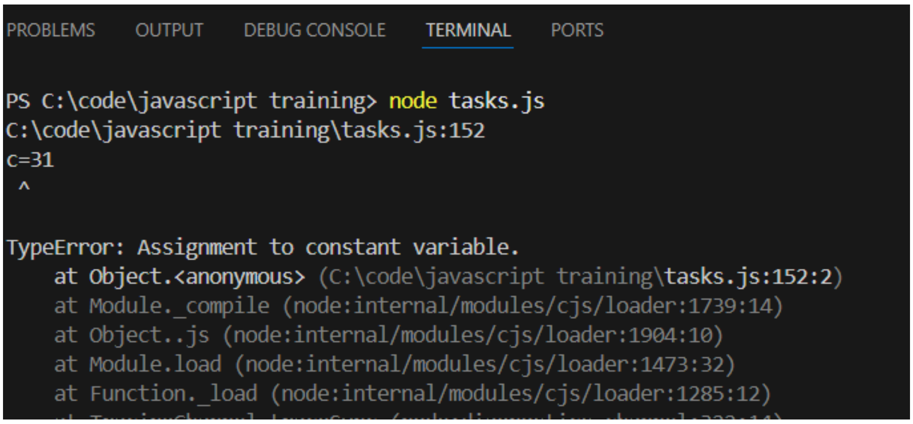
c=31

console.log(a);

console.log(b);

console.log(c);

Output:



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:152
c=31
 ^

TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\code\javascript training\tasks.js:152:2)
    at Module._compile (node:internal/modules/cjs/loader:1739:14)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
```

TASK 17:

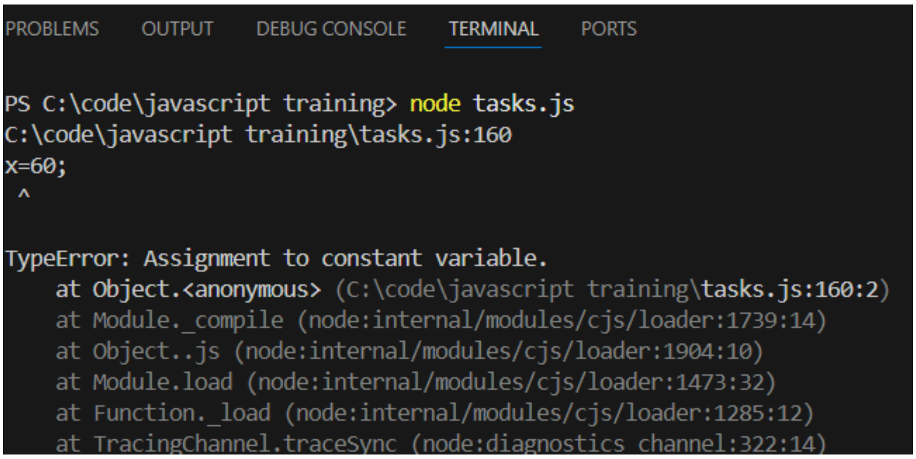
Code:

const x=50;

x=60;

console.log(x);

Output:



```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:160
x=60;
 ^

TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\code\javascript training\tasks.js:160:2)
    at Module._compile (node:internal/modules/cjs/loader:1739:14)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
```

TASK 18:

Code:

let m;

console.log(m);

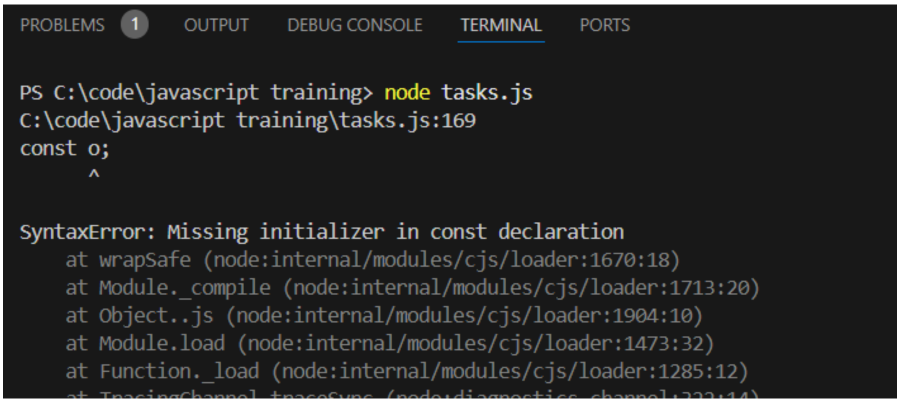
var n;

console.log(n);

const o;

console.log(o);

Output:



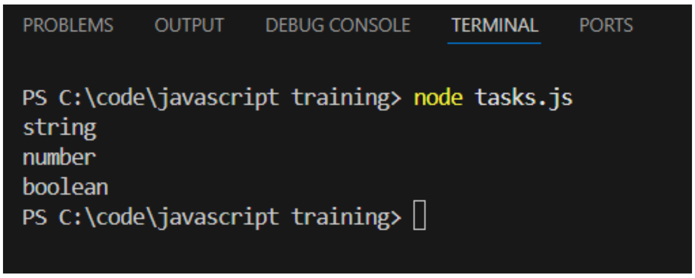
```
PS C:\code\javascript training> node tasks.js
C:\code\javascript training\tasks.js:169
const o;
      ^

SyntaxError: Missing initializer in const declaration
    at wrapSafe (node:internal/modules/cjs/loader:1670:18)
    at Module._compile (node:internal/modules/cjs/loader:1713:20)
    at Object..js (node:internal/modules/cjs/loader:1904:10)
    at Module.load (node:internal/modules/cjs/loader:1473:32)
    at Function._load (node:internal/modules/cjs/loader:1285:12)
    at TracingChannel.traceSync (node:diagnostics_channel:322:14)
```

TASK 19:

Code:

```
let p="Kavi";

let q=20;

let r=true;

console.log(typeof(p));

console.log(typeof(q));

console.log(typeof(r));
```

Output:



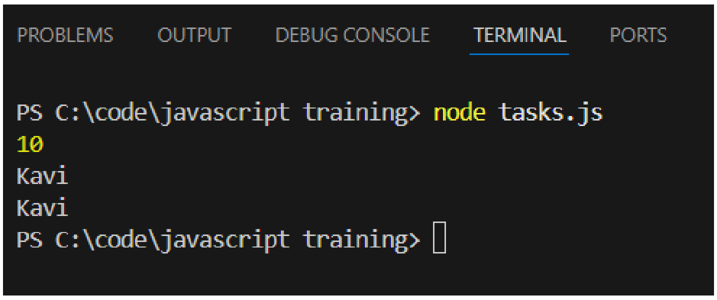
```
PS C:\code\javascript training> node tasks.js
string
number
boolean
PS C:\code\javascript training> 
```

TASK 20:

Code:

```
let x=10;

let y=10;

console.log(x);

function user(){

     let name="Kavi";

    console.log(name);

}

function user(){

     let fullname="Kavi";

    console.log(fullname);

}

user(user.name);

user(user.fullname);
```

Output:



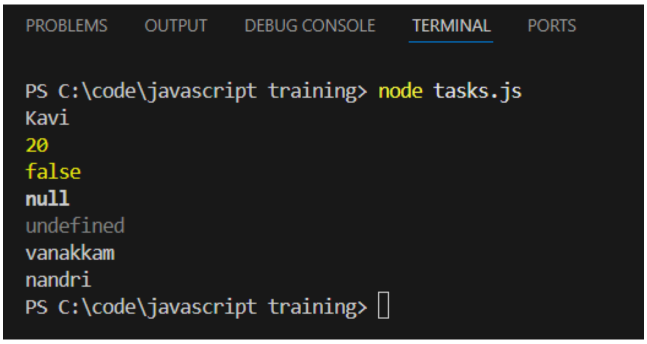
```
PS C:\code\javascript training> node tasks.js
10
Kavi
Kavi
PS C:\code\javascript training> 
```

TASK 21:

Code:

```
let name="Kavi";

let age=20;

let isMarried=false;

let children=null;

let address;

let greets={

   first:"vanakkam",

   last:"nandri"

};

console.log(name);

console.log(age);

console.log(isMarried);

console.log(children);

console.log(address);

console.log(greets.first);

console.log(greets.last);
```

Output:



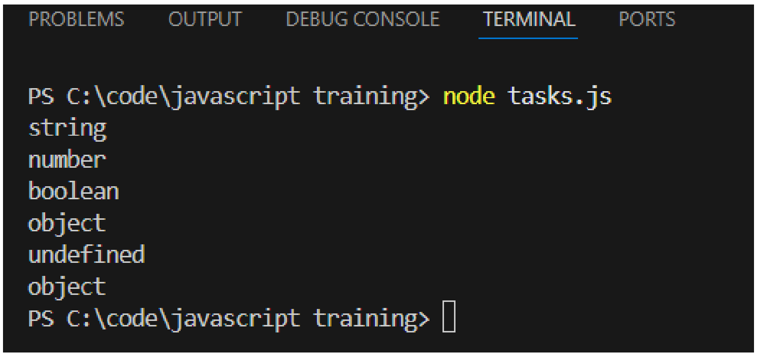
```
PS C:\code\javascript training> node tasks.js
Kavi
20
false
null
undefined
vanakkam
nandri
PS C:\code\javascript training> 
```

TASK 22:

Code:

```
let name="Kavi";

let age=20;

let isMarried=false;

let children=null;

let address;

let greets={

   first:"vanakkam",

   last:"nandri"

};

console.log(typeof(name));

console.log(typeof(age));

console.log(typeof(isMarried));

console.log(typeof(children));

console.log(typeof(address));

console.log(typeof(greets));
```

Output:



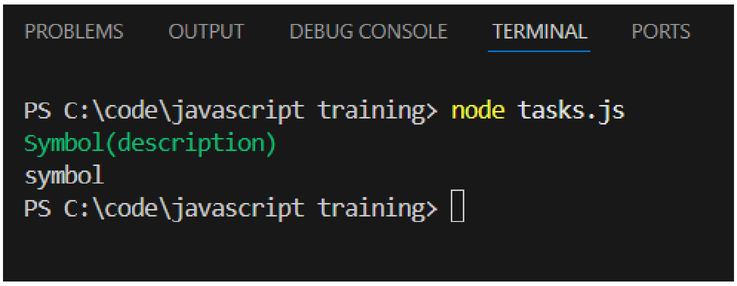
```
PS C:\code\javascript training> node tasks.js
string
number
boolean
object
undefined
object
PS C:\code\javascript training> 
```

TASK 23:

Code:

```
const mysymbol=Symbol('description');

console.log(mysymbol);

console.log(typeof(mysymbol));
```

Output:



```
PS C:\code\javascript training> node tasks.js
Symbol(description)
symbol
PS C:\code\javascript training> 
```

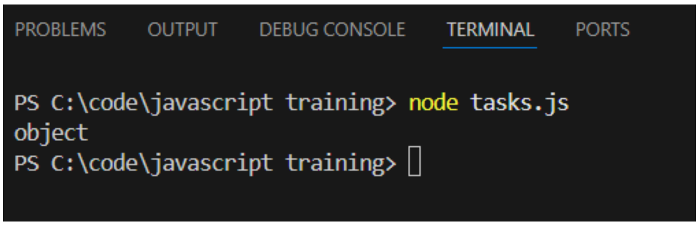
TASK 24:

Code:

let actor=null;

console.log(typeof(actor));

Output:



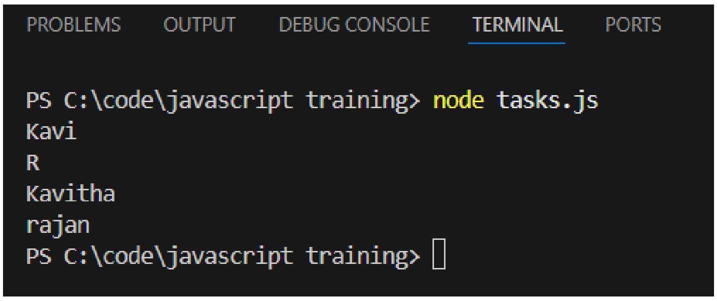
```
PS C:\code\javascript training> node tasks.js
object
PS C:\code\javascript training> 
```

TASK 25:

Code:

```
let fullname="Kavi";
var surname="R";
function user(){
    let fullname="Kavitha";
    var surname="rajan";
   console.log(fullname);
   console.log(surname);
}
console.log(fullname);
console.log(surname);
user();
```

Output:



```
PS C:\code\javascript training> node tasks.js
Kavi
R
Kavitha
rajan
PS C:\code\javascript training> 
```

TASK 26:

Code:

//explicit

let name1="Kavi";

let name2="123";

let namenum11=Number(name1);

console.log(typeof(namenum11));

console.log(namenum11);

let namenum21=Number.parseInt(name1);

console.log(typeof(namenum21));

console.log(namenum21);

//implicit

let namenum31=name1*1;

console.log(typeof(namenum31));

console.log(namenum31);

let namenum12=Number(name2);

console.log(typeof(namenum12));

console.log(namenum12);

let namenum22=Number.parseInt(name2);

console.log(typeof(namenum22));

console.log(namenum22);

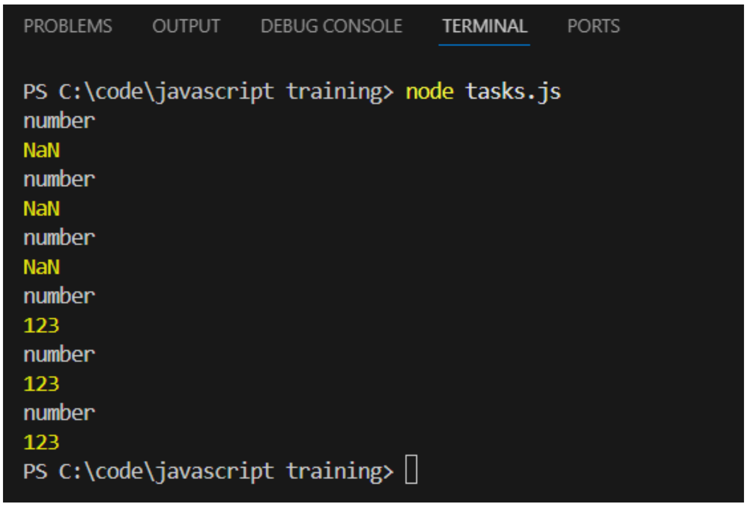
//implicit

let namenum32=name2*1;

console.log(typeof(namenum32));

console.log(namenum32);

Output:



```
PS C:\code\javascript training> node tasks.js
number
NaN
number
NaN
number
NaN
number
123
number
123
number
123
PS C:\code\javascript training> 
```

TASK 27:

Code:

let name="Kavi";

let isMarried=false;

let namebool=Boolean(name);

let isMarriedstr=String(isMarried);

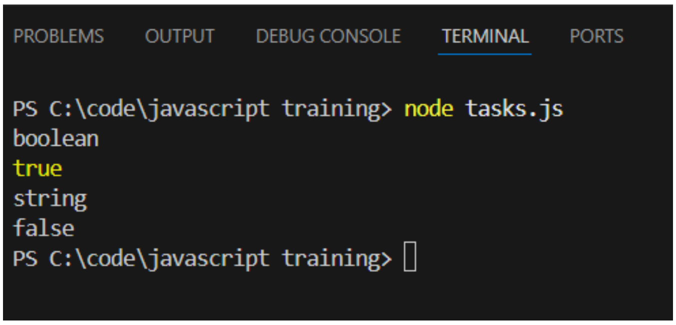
console.log(typeof(namebool));

console.log(namebool);

console.log(typeof(isMarriedstr));

console.log(isMarriedstr);

Output:



```
PS C:\code\javascript training> node tasks.js
boolean
true
string
false
PS C:\code\javascript training> 
```

TASK 28:

Code:

let x=10;

let y=20;

let z=x+y;

let a=y-x;

let b=x*2;

let c=y/a;

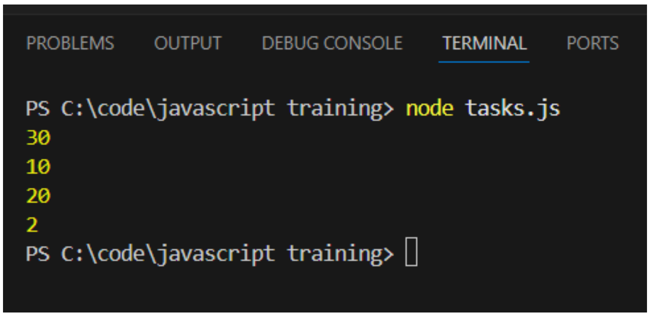
console.log(z);

console.log(a);

console.log(b);

console.log(c);

Output:



```
PS C:\code\javascript training> node tasks.js
30
10
20
2
PS C:\code\javascript training> 
```

TASK 29:

Code:

let k=5;

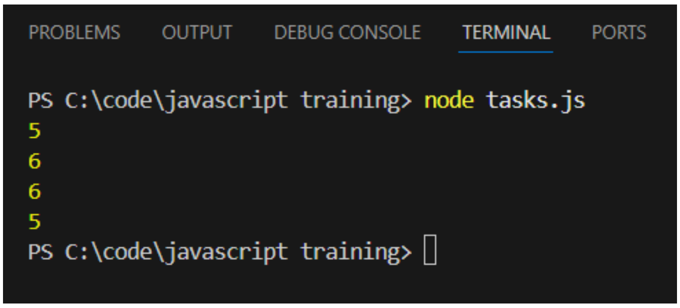
console.log(k++);

console.log(k--);

console.log(++k);

console.log(--k);

Output:



```
PS C:\code\javascript training> node tasks.js
5
6
6
5
PS C:\code\javascript training> 
```

TASK 30:

Code:

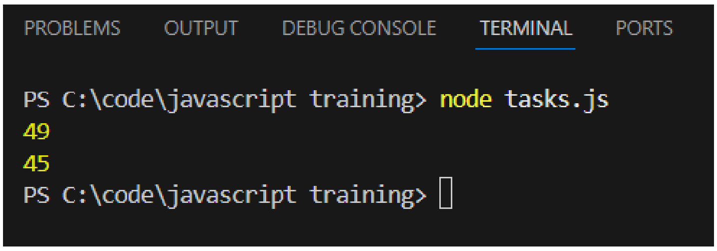
let result1=10*5-4+6/2;

console.log(result1);

let result2=(10*5)-(4+6)/2;

console.log(result2);

Output:



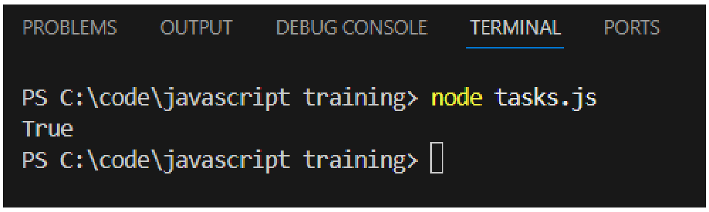
```
PS C:\code\javascript training> node tasks.js
49
45
PS C:\code\javascript training> 
```

TASK 31:

Code:

```
let p=5;

let q=10;

let r=5;

if(p<=r){

  console.log("True");

}
```

Output:



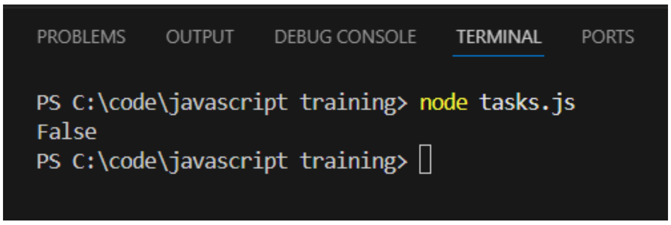
```
PS C:\code\javascript training> node tasks.js
True
PS C:\code\javascript training> 
```

TASK 32:

Code:

```
let a=10;
let b="Ten";
let c=null;
let d=20;
let e=0;
if( e===c || a===e || a==b || a==d){
   console.log("True");
}
else{
   console.log("False");
}
```

Output:



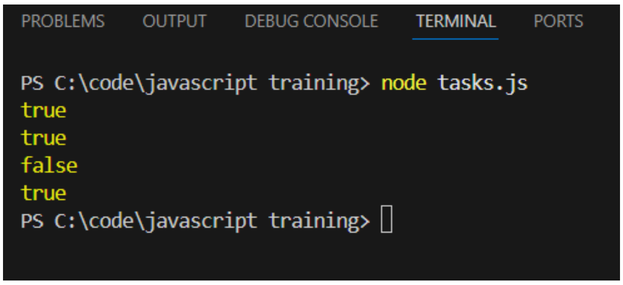
```
PS C:\code\javascript training> node tasks.js
False
PS C:\code\javascript training> 
```

TASK 33:

Code:

```
let a="apple";
let b="banana";
let c="amla";
console.log(a<b);
console.log(c<a);
console.log(a===c);
console.log("cat"<"cub");
```

Output:



```
PS C:\code\javascript training> node tasks.js
true
true
false
true
PS C:\code\javascript training> 
```

TASK 34:

Code:

let a="apple";

let b="banana";

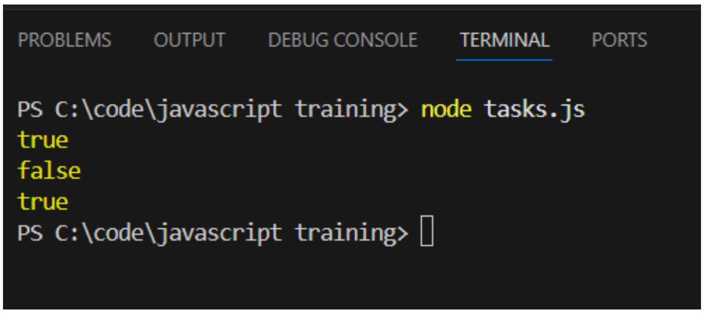
let c="apple";

console.log(a!=b);

console.log(c!==a);

console.log("cat"!=="cub");

Output:



```
PS C:\code\javascript training> node tasks.js
true
false
true
PS C:\code\javascript training> 
```

TASK 35:

Code:

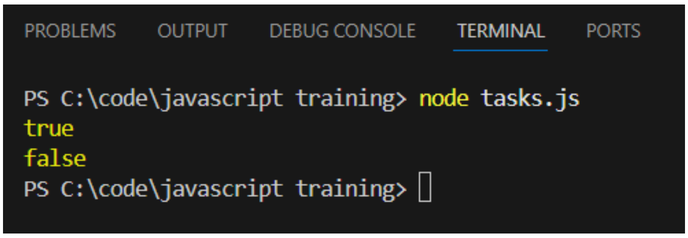
let a=null;

let b;

console.log(a==b);

console.log(a===b);

Output:



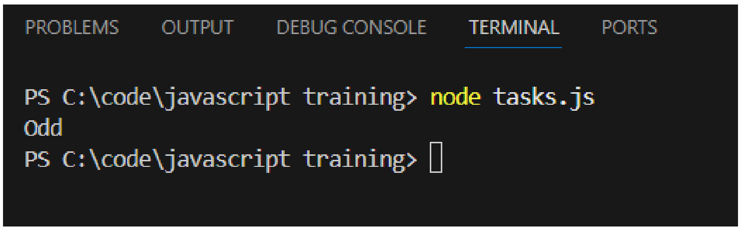
```
PS C:\code\javascript training> node tasks.js
true
false
PS C:\code\javascript training> 
```

TASK 36:

Code:

```
let a=11;
if(a%2==0){
   console.log("Even");
}
else if(a%2==1){
   console.log("Odd");
}
else{
   console.log("Neither");
}
```

Output:



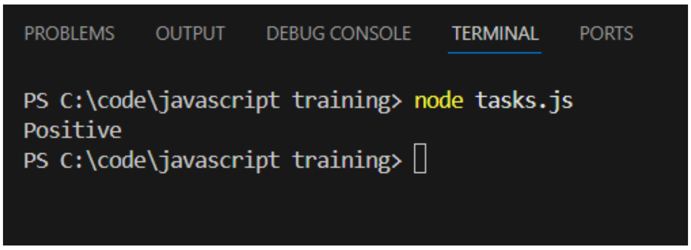
```
PS C:\code\javascript training> node tasks.js
Odd
PS C:\code\javascript training> 
```

TASK 37:

Code:

```
let a=123;
if(a>0){
   console.log("Positive");
}
else if(a<0){
   console.log("Negative");
}
else if(a==0){
   console.log("Zero");
}
else{
   console.log("Neither");
}
```

Output:



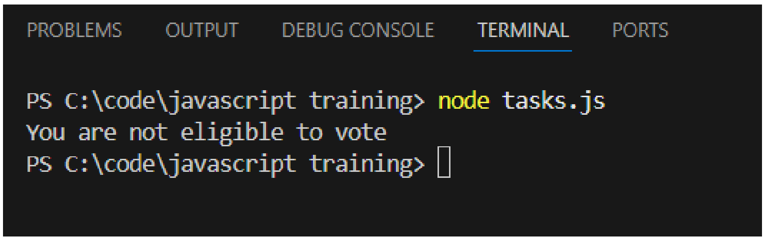
```
PS C:\code\javascript training> node tasks.js
Positive
PS C:\code\javascript training> 
```

TASK 38:

Code:

```
let age=17;
console.log((age>18)?"You are eligible to vote":"You are not eligible to vote");
```

Output:



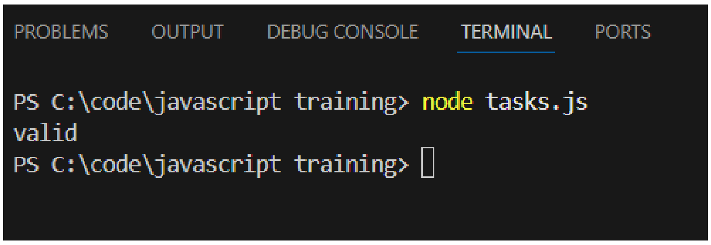
```
PS C:\code\javascript training> node tasks.js
You are not eligible to vote
PS C:\code\javascript training> 
```

TASK 39:

Code:

```
let marks=90;
let percentage=(marks/100)*100;
let result=(percentage>70)?"pass":"fail";
if(result=="pass"){
   console.log("valid");
}else if(result=="fail"){
    console.log("invalid");
}else{
    console.log("Neither");
}
```

Output:



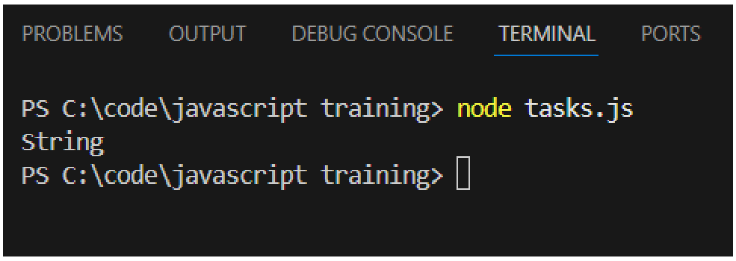
```
PS C:\code\javascript training> node tasks.js
valid
PS C:\code\javascript training> 
```

TASK 40:

Code:

```
let a="Muruga";
if((typeof(a)=="string")?atype="String":atype="Neither");
console.log(atype);
```

Output:



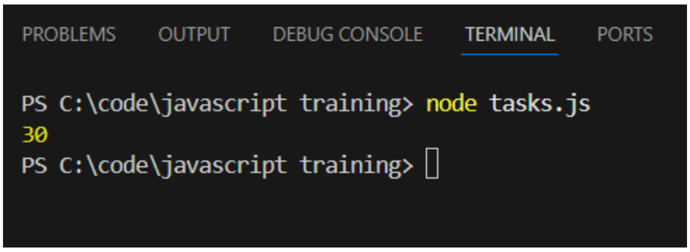
```
PS C:\code\javascript training> node tasks.js
String
PS C:\code\javascript training> 
```

TASK 41:

Code:

```
let a = 10;
let b = 20;
let c=a+b;
if(a>b || a==b && a!=0){
    console.log(a);
}else if(!(c<a) && c%2==0 ){
 console.log(c);
}
```

Output:



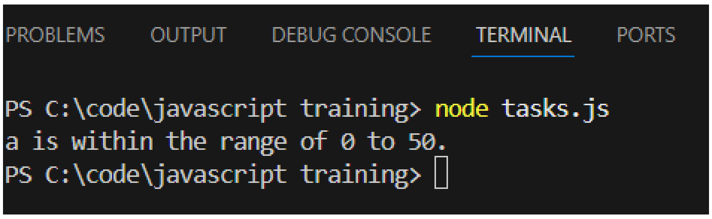
```
PS C:\code\javascript training> node tasks.js
30
PS C:\code\javascript training> 
```

TASK 42:

Code:

```
let a=20;

if(!(a<0) && !(a>50)){

   console.log("a is within the range of 0 to 50.");

}
```

Output:



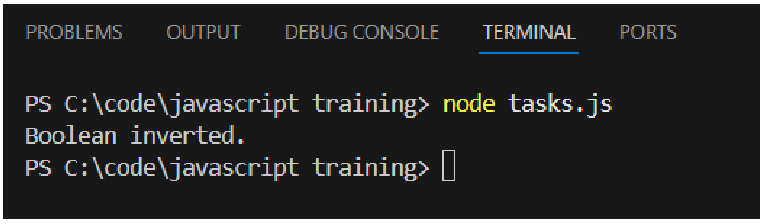
```
PS C:\code\javascript training> node tasks.js
a is within the range of 0 to 50.
PS C:\code\javascript training> 
```

TASK 43:

Code:

```
let d=false;

if(!d==true){

   console.log("Boolean inverted.");

}
```

Output:



```
PS C:\code\javascript training> node tasks.js
Boolean inverted.
PS C:\code\javascript training> 
```

TASK 44:

Code:

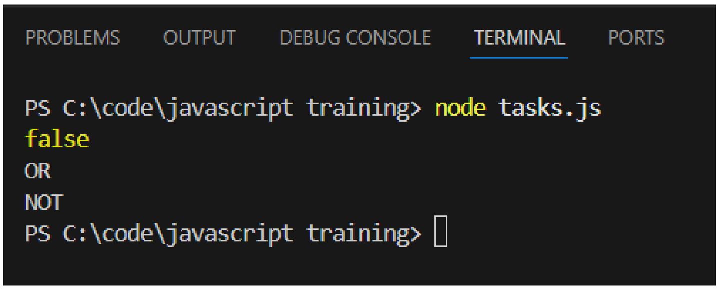
```
let isready=false;
```

console.log(isready && "AND");

console.log(isready || "OR");

console.log(!isready && "NOT");

Output:



```
PS C:\code\javascript training> node tasks.js
false
OR
NOT
PS C:\code\javascript training> 
```

TASK 45:

Code:

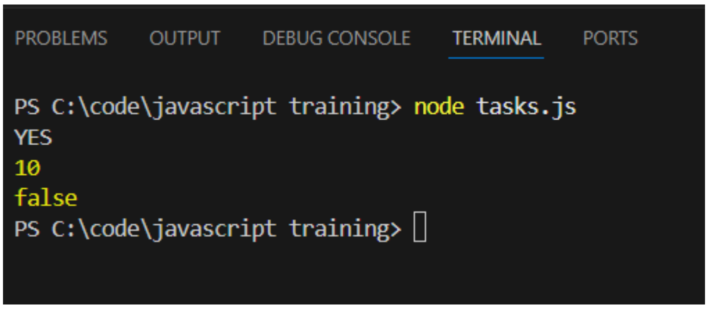
let isready=10;

console.log(isready && "YES");

console.log(isready || "YES");

console.log(!isready && "Yes");

Output:



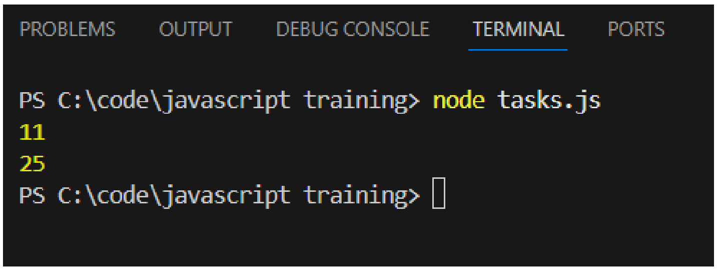
```
PS C:\code\javascript training> node tasks.js
YES
10
false
PS C:\code\javascript training> 
```

TASK 46:

Code:

```
function add(a, b){
   let c=a+b;
   console.log(c);
}add(5,6);
add(10,15);
```

Output:



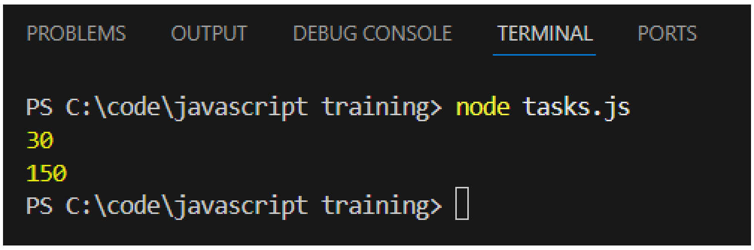
```
PS C:\code\javascript training> node tasks.js
11
25
PS C:\code\javascript training> 
```

TASK 47:

Code:

```
function area(l, b){
   let a=l*b;
   console.log(a);
}
area(5,6);
area(10,15);
```

Output:



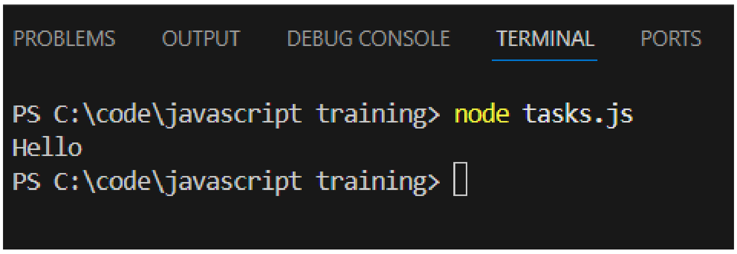
```
PS C:\code\javascript training> node tasks.js
30
150
PS C:\code\javascript training> 
```

TASK 48:

Code:

```
function greets(){
   console.log("Hello");
}
greets();
```

Output:



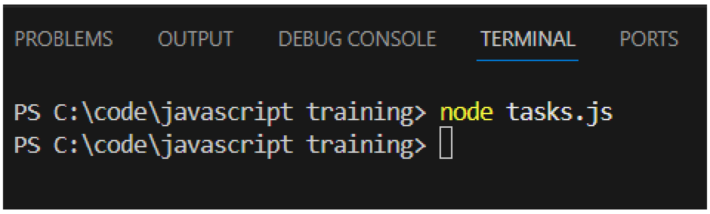
```
PS C:\code\javascript training> node tasks.js
Hello
PS C:\code\javascript training> 
```

TASK 49:

Code:

```
function nothing(){

}

nothing();
```

Output:



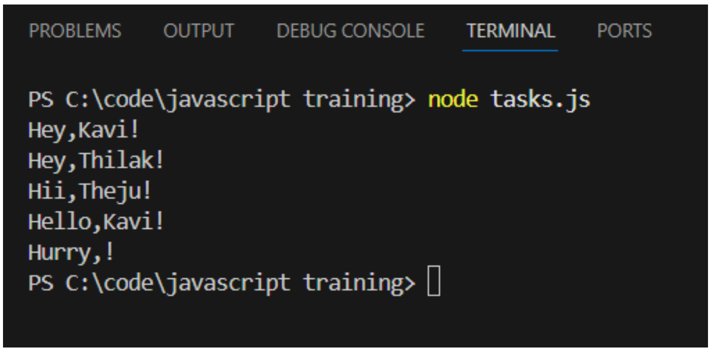
```
PS C:\code\javascript training> node tasks.js
PS C:\code\javascript training> 
```

TASK 50:

Code:

```
function greets(name="Kavi",greetings="Hey"){

   console.log(`${greetings},${name}!`);

}

greets();

greets("Thilak");

greets("Theju","Hii");

greets(undefined,"Hello");

greets("","Hurry");
```

Output:



```
PS C:\code\javascript training> node tasks.js
Hey,Kavi!
Hey,Thilak!
Hii,Theju!
Hello,Kavi!
Hurry,!
PS C:\code\javascript training> 
```

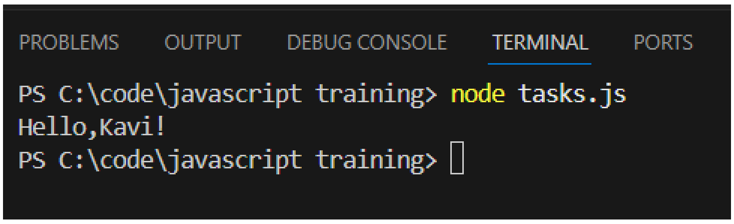
TASK 51:

Code:

```
let greet=(name)=>console.log(`Hello,${name}!`);
```

greet('Kavi');

Output:



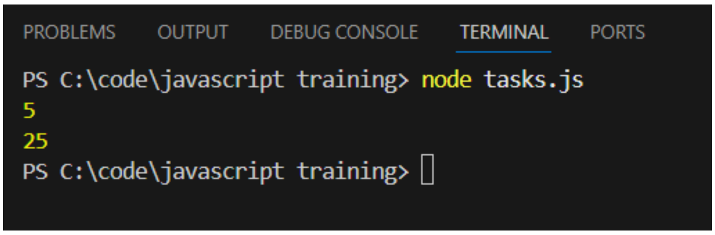
```
PS C:\code\javascript training> node tasks.js
Hello,Kavi!
PS C:\code\javascript training> 
```

TASK 52:

Code:

```
let add=(a,b)=>console.log(a+b);

add(2,3);

add(10,15);
```

Output:



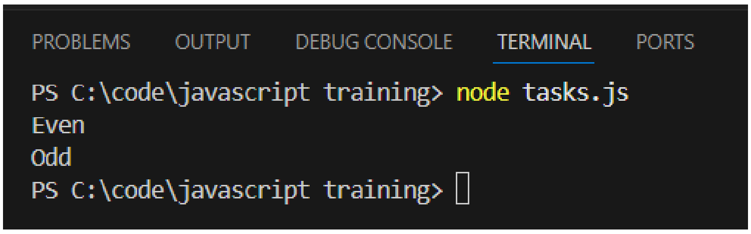
```
PS C:\code\javascript training> node tasks.js
5
25
PS C:\code\javascript training> 
```

TASK 53:

Code:

```
let isEven=(num)=>{
    if(num%2==0){
        console.log("Even");
    }else if(num%2==1){
        console.log("Odd");
    }else{
        console.log("Invalid");
    }
}
isEven(20);
isEven(3);
```

Output:



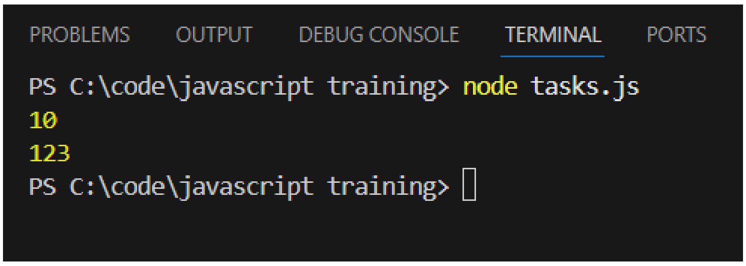
```
PS C:\code\javascript training> node tasks.js
Even
Odd
PS C:\code\javascript training> 
```

TASK 54:

Code:

```
let maxValue=(a,b)=>{
   if(a>b){
      return(a);
   }else if(b>a){
      return(b);
   }
}
console.log(maxValue(5,10));
console.log(maxValue(123,121));
```

Output:



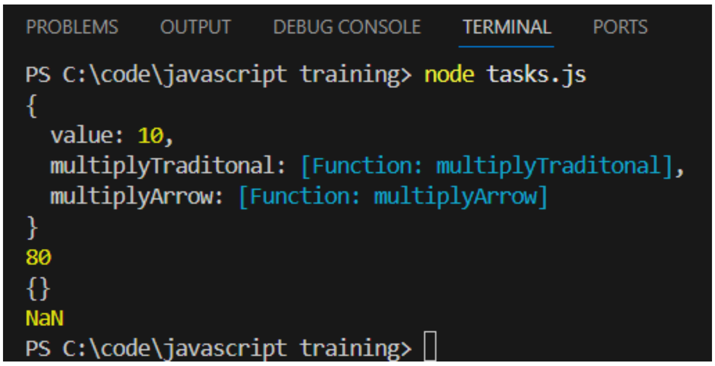
```
PS C:\code\javascript training> node tasks.js
10
123
PS C:\code\javascript training> 
```

TASK 55:

Code:

```
const myObject = {
   value:10,
   multiplyTraditonal(n){
     console.log(this);
     return this.value *n;
   },
  multiplyArrow: (n)=>{
    console.log(this);
     return this.value*n;
  }
};
console.log(myObject.multiplyTraditonal(8));
console.log(myObject.multiplyArrow(11));
```

Output:



```
PS C:\code\javascript training> node tasks.js
{
  value: 10,
  multiplyTraditonal: [Function: multiplyTraditonal],
  multiplyArrow: [Function: multiplyArrow]
}
80
{}
NaN
PS C:\code\javascript training> 
```