

## Code 1

```
let x;  
var a = 5;  
var b = 10;  
var c = function(a,b,c){  
    document.write(x); // undefined  
    document.write(a); // 8  
    var f = function(a, b, c){  
        b = a;  
        document.write(b); // 8  
        b = c;  
        var x = 5;  
    }  
    f(a, b, c);  
    document.write(b); // 9  
    var x = 10;  
}  
c(8, 9, 10);  
document.write(b); // 10  
document.write(x); // undefined
```

## Output Code1

```
undefined88910undefined
```

## Code 2

```
var x = 9;  
function myFunction(){  
    return x * x;  
}  
document.write(myFunction()) // 81  
x = 5;  
document.write(myFunction()) // 25
```

## Output Code2

```
8125
```

## Code 3

```
var foo = 1;  
function bar(){  
    if(!foo){ // foo = undefined  
        var foo = 10;  
    }  
    alert(foo); // 10  
}  
bar()
```

## Output Code3

10

Problem 2: Draw the lexical environment diagram

### Code 1

```
let x;  
var a = 5;  
var b = 10;  
var c = function(a,b,c){  
  document.write(x); // undefined  
  document.write(a); // 8  
  var f = function(a, b, c){  
    b = a;  
    document.write(b); // 8  
    b = c;  
    var x = 5;  
  }  
  f(a, b, c);  
  document.write(b); // 9  
  var x = 10;  
}  
c(8, 9, 10);  
document.write(b); // 10  
document.write(x); // undefined
```

### Lexical Environment Diagram For Code 1

```
1. Global EC creation:  
Outer: null, this: Window, LE: [{a: undefined, b: undefined, c: undefined}],  
TDZ{x}  
2. Global EC execution:  
Outer: null, this: Window, LE: [{a: 5, b: 10, c: fn, x: undefined}], TDZ{}  
3. c FEC creation:  
Outer: global, this: ..., LE: [{arguments: obj, a: 8, b: 9, c: 10, f:  
undefined, x: undefined}], TDZ{}  
4. c FEC execution:  
Outer: global, this: ..., LE: [{arguments: obj, a: 8, b: 9, c: 10, f: fn, x:  
10}], TDZ{}  
5. f FEC creation:  
Outer: c function block, this: ..., LE: [{arguments: obj, a: 8, b: 9, c: 10,  
x: undefined}], TDZ{}  
6. f FEC execution:  
Outer: c function block, this: ..., LE: [{arguments: obj, a: 8, b: 10, c:  
10, x: undefined 5}], TDZ{}
```

### Code 2

```
var x = 9;  
function myFunction(){  
  return x * x;
```

```

}
document.write(myFunction()) // 81
x = 5;
document.write(myFunction()) // 25

```

## Lexical Environment Diagram For Code 2

```

1. Global EC creation:
Outer: null, this: Window, LE: [{x: undefined, myFunction: fn}], TDZ{}
2. Global EC execution:
Outer: null, this: Window, LE: [{x: 9, myFunction: fn}], TDZ{}
3. myFunction FEC creation:
Outer: global, this: ..., LE: [{arguments: obj}], TDZ{}
4. myFunction FEC execution:
Outer: global, this: ..., LE: [{arguments: obj}], TDZ{}
5. Global EC execution:
Outer: null, this: Window, LE: [{x: 5, myFunction: fn}], TDZ{}
7. myFunction FEC creation:
Outer: global, this: ..., LE: [{arguments: obj}], TDZ{}
8. myFunction FEC execution:
Outer: global, this: ..., LE: [{arguments: obj}], TDZ{}

```

## Code 3

```

var foo = 1;
function bar(){
  if(!foo){ // foo = undefined
    var foo = 10;
  }
  alert(foo); // 10
}
bar()

```

## Lexical Environment Diagram For Code 3

```

1. Global EC creation:
Outer: null, this: Window, LE: [{foo: undefined, bar: fn}], TDZ{}
2. Global EC execution:
Outer: null, this: Window, LE: [{foo: 1, bar: fn}], TDZ{}
3. bar FEC creation:
Outer: global, this: ..., LE: [{arguments: obj, foo: undefined}], TDZ{}
4. myFunction FEC execution:
Outer: global, this: ..., LE: [{arguments: obj}, foo: 10], TDZ{}

```