

HOMEWORK 1

PURVA THAKKAR

B00743825

//Problem 1

```
function selectListsWithLengthMultiple(list, mul=2) {
  return list.reduce(function(acc, cur) {
    cur.length % mul === 0 ? acc.push(cur) : null
    return acc
  }, []);
}
```

//Problem 2

```
function reverse(text) {
  return text.split("").reverse().join("");
}
```

//Problem 3

```
function isPalindrome(str) {
  return
  (str.replace(/\W+/g, "").toLowerCase()) === (str.replace(/\W+/g, "").toLowerCase().split("").reverse().join(""));
}
```

//Problem 4

```
function minLenArg(...argList) {
  return argList.reduce(function(acc, cur) {
    cur.length === Math.min(...argList.map(el => el.length)) ? acc.push(cur) : null
    return acc
  }, [])[0];
};
```

//Problem 5

```
function formatInt(x, {base, n, sep}){
  return x.toString(base).split("").reverse().join("").match(new RegExp('.{1,' + n + '}',
  'g')).join(sep).split("").reverse().join("");
}
```

//Problem 6

```
function isEvenParity(x){
  return ((x.toString(2).match(/1/g).length)%2 == 0) ? true : false;
}
```

//Problem 7

```
function bitIndexesToInt(list) {
  return parseInt(list.length === 0 ? [0] : Array.from({length: Math.max(...list)+1}, (v, i) =>
  i).reverse().reduce((acc, cur) => {
    list.includes(cur) ? acc += '1' : acc += '0'
    return acc
  }, ""), 2);
};
```

PURVA THAKKAR

Homework 1

B00743825

HOMEWORK 1

PURVA THAKKAR

B00743825

//Problem 8

```
function intToBitIndexes(int) {  
  return int.toString(2).split("").reverse().reduce((acc, cur, index) => {  
    cur === '1' ? acc.push(index) : null  
    return acc  
  }, []);  
}
```

//Problem 9

```
function multiIndex(obj, indexes){  
  return indexes.split(".").filter(v => v).reduce((accumulator, currentValue) => accumulator[currentValue],  
  obj );  
}
```

//Problem 10

```
function zip(list1, list2) {  
  return list1.map((e, i) => [e, list2[i]]);  
}
```

//Problem 11

```
function multiZip (array, ...arrays) {  
  return array.map((e, i) => arrays.reduce((a, array) => [...a, array[i]], [e]));  
}
```

//Problem 12

```
function multiZipAny(array, ...arrays) {  
  return array.map((e, i) => arrays.reduce((a, array) => [...a, array[i]],  
  [e])).slice(0,arguments[Math.min(arrays.length)].length);  
}
```

//Problem 13

The use of == operator is not advisable here as it does type coercion thus automatically converting result[key] to the type null in order to carry out the comparison. Instead, === operator should be used which checks for both type and value for comparison.

Replacement:

```
const results = ...;  
if (results[key] === null) {  
  console.log(`${key} is not defined`);  
}
```

//Problem 14

Various Design Patterns described in the GOF book are criticized and can be replaced.

1. Singleton Pattern can be replaced by MetaClasses
2. Interpreter Pattern can be replaced by Macros (extending the language) EvalFunction, MetaCircularInterpreter Support for parser generation (for differing syntax)
3. Visitor Pattern can be replaced by GenericFunctions (MultipleDispatch)
4. Iterator Pattern can be replaced by AnonymousFunctions (used with HigherOrderFunctions, MapFunction, FilterFunction, etc.)

HOMEWORK 1

PURVA THAKKAR

B00743825

5. Factory Pattern can be replaced by MetaClasses, closures

//Problem 15

a) when using an arrow function this is not bound to anything and it just inherits it from the parent scope. In this case we need this keyword in our distance() and area() functions that's why we are using regular function

b)

```
function Shape(x, y) { this.x = x; this.y = y; }
```

```
Shape.prototype.distance = function(other) {  
  console.log(this.x, this.y, other)  
  return Math.sqrt((other.x - this.x)**2 +  
    (other.y - this.y)**2);  
}
```

```
function Rect(x, y, w, h) {  
  Shape.call(this, x, y); //==> calling variables without using super  
  this.width = w; this.height = h;  
}  
Rect.prototype = Object.create(Shape.prototype);
```

```
Rect.prototype.constructor = Rect;
```

```
Rect.prototype.area = function() {  
  console.log(this.width, this.height);  
  return this.width * this.height;  
};
```