ALGORITHM - II

By Divya Maheswaran

Code Sample 1 - function and return I

```
1) function a(){
2)     console.log('hello');
3) }
4) console.log('Dojo');
```

OUTPUT:

Dojo

Code Sample 2 - function and return I

```
1) function a(){
2)   console.log('hello');
3)   return 15;
4) }
5) x = a();
6) console.log('x is', x);
```

OUTPUT:

Hello

X is 15

Code Sample 3 - function and return I

```
1) function a(n){
2)   console.log('n is', n);
3)   return n+15;
4) }
5) x = a(3);
6) console.log('x is', x);
```

OUTPUT:

N is 3

X is 18

Code Sample 4 - function and return I

```
1)  function a(n){
2)    console.log('n is', n);
3)    y = n*2;
4)    return y;
5)  }
6)  x = a(3) + a(5);
7)  console.log('x is', x);
```

OUTPUT:

N is 3

N is 5

X is 16

Code Sample 5 - order of operations I

```
1)  function op(a,b){
2)    c = a+b;
3)    console.log('c is', c);
4)    return c;
5)  }
6)  x = op(2,3) + op(3,5);
7)  console.log('x is', x);
```

OUTPUT:

C is 5

C is 8

X is 13

Code Sample 6 - order of operations II

```
1)  function op(a,b){
2)    c = a+b;
3)    console.log('c is', c);
4)    return c;
5)  }
6)  x = op(2,3) + op(3,op(2,1)) + op(op(2,1),op(2,3));
7)  console.log('x is', x)
```

OUTPUT:

C is 5

C is 3

C is 6

C is 3

C is 5

C is 8

X is 19

Code Sample 7 - scoping

```
1) var x = 15;
2) function a(){
3)    var x = 10;
4) }
5) console.log(x);
6) a();
7) console.log(x);
```

OUTPUT:

15

15

```
function multiply(x,y){
    console.log(x);
    console.log(y);
}
b = multiply(2,3);
console.log(b);
```

OUTPUT:

2

3

undefined

```
function multiply(x,y){
    return x*y;
}
b = multiply(2,3);
console.log(b);
console.log(multiply(5,2));
```

OUTPUT:

6

10

```
var x = [1,2,3,4,5,10];
for(var i=0; i<5; i++)
{
   i = i + 3;
   console.log(i);
}
var x=15;
console.log(x);
```

OUTPUT:

3

4

15

```
function awesome(){
    var x=10;
    console.log(x);
}
console.log(x);
awesome();
console.log(x);
```

OUTPUT:

error

```
for(var i=0; i<15; i+=2){
   console.log(i);
}
```

OUTPUT:

0

2

4

6

8

10

14

```
for(var i=0; i<3; i++){
    for(var j=0; j<2; j++){
        console.log(i*j);
    }
}
```

OUTPUT:

0

0

0

1

0

2


```
function looping(x,y){
    for(var i=0; i<x; i++){
        for(var j=0; j<x; j++){
            console.log(i*j);
        }
    }
}
z = looping(3,3);
console.log(z);
```

OUTPUT:

0

0

0

0

1

2

0

2

4

undefined


```
function looping(x,y){
    for(var i=0; i<x; i++){
        for(var j=0; j<y; j++){
            console.log(i*j);
        }
    }
    return x*y;
}
z = looping(3,5);
```

```
console.log(z);
```

OUTPUT:

0

0

0

0

0

0

1

2

3

4

0

2

4

6

8

15

Part 2

Print 1 to x

Please complete the codes below to have the function print all the integers from 1 to x. If x is

negative, have it print/log "negative number" and have the function return false.

```
function printUpTo(x){
  if(x<0){
      console.log("negative number");
      return false;
      }
    for(var i=1; i<=x; i++){
      console.log(i);
      }
}
printUpTo(1000); // should print all the integers from 1 to 1000
y = printUpTo(-10); // should return false
console.log(y); // should print false
```

PrintSum

Please complete the code below to have the function print integers from 0 to x and with each integer

print the sum so far. Have the function return the final sum

```
function printSum(x){
   var sum = 0;
   for(var i=1; i<=x; i++){
      sum = sum+i;
      console.log("Number:", i, "Sum:", sum);
      }
   return sum;
}
y = printSum(255);
console.log(y);
```

PrintSumArray

Please complete the code below to have the function return the sum of all the values in a given array

```
function printSumArray(x){
   var sum = 0;
   for(var i=0; i<x.length; i++) {
     sum = sum+x[i];
   }
   return sum;
}
console.log( printSumArray([1,2,3]) ); // should log 6
```

Bonus: LargestElement

Create a function that returns the largest element in a given array. For example largestElement([1,30,5,7]) should return 30

```
function largestElement(x){
  var max=x[0];
  for(var i=0; i<x.length; i++){
    if(x[i]> max){
      max=x[i];
    }
  }
  console.log(max);
}
largestElement( [1,30,5,7] );
```