```
1. function nestedMultiplicative(n) {
   let count = 0;
   for (let i = 1; i < n; i *= 2) {
      for (let j = 1; j < n; j++) {
         count++;
      }
   }
   return count;
}
```

```
2. function dependentNestedLoop(n) {
   let count = 0;
   for (let i = 0; i < n; i++) {
      for (let j = i; j < n; j++) {
         for (let k = j; k < n; k++) {
            count++;
         }
      }
   }
   return count;
}
```

```
3.
function decreasingLoop(n) {
   let count = 0;
   for (let i = n; i > 0; i = Math.floor(i / 2)) {
      for (let j = 0; j < i; j++) {
         count++;
      }
   }
   return count;
}
```

```
4.function primeCheckLoop(n) {
   let count = 0;
   for (let i = 2; i <= n; i++) {
      let isPrime = true;
      for (let j = 2; j * j <= i; j++) {
         if (i % j === 0) {
            isPrime = false;
```

```
                break;
            }
        }
        if (isPrime) count++;
    }
    return count;
}
```

```
5.function logarithmicInnerLoop(n) {
    let count = 0;
    for (let i = 0; i < n; i++) {
        for (let j = 1; j < n; j *= 2) {
            count++;
        }
    }
    return count;
}
```

6.

```
for(var i=0;i<10;i++)
{
   for(var j=0;j<N;j++)
       {
           for(var k=N-2;k<N+2;k++)
              {
                 console.log(i,j);
              }
       }
}
```