## 面试测试题(以下试题涉及使用程序实现时，可选用任意一种熟悉的流行语言)

1. 简述以下对”堆栈”和”队列”的理解，给出示意简图，并用代码实现。
2. 用代码实现十进制转三十二进制，并给出测试代码。
3. 首先看以下代码段

```JavaScript

var a = ‘abc’, b = ‘123’, c = [abc, 123];

var arr = [a, b, c];

var newArr = arr;

c = ‘abc123’;

console.log(newArr);

```

给出你认为的输出，并简述一下理由。

1. 我们来看以下代码段

```javascript

for (var i = 0;i < 5;i++) {

setTimeout(function(){

console.log( ‘i: ’, i);

}, 0)

}

console.log(i)

```

请给出你的输出结果，并简述理由。

如若稍作修改，将代码重构如下：

```javascript

for (var i = 0;i < 5;i++) {

setTimeout(function(){

console.log( ‘2: ’, i);

}, 0)

console.log(‘1:’, i) //这里输出想要的结果

}

console.log(i)

```

请给出你的输出结果，并简述理由。

有没有想过遇到类似的情况，该怎么处理呢？如何才能避免这种情况，拿到你想要的结 果呢？简单来说就是在第一段代码的基础上稍作修改输出第二段代码注释前的结果。

好了，接下来让我们发散以下思维，看一下下面这段代码

```javascript

var a = new Promise((resolve, reject) => {

console.log(1);

setTimeout(() => {console.log(2)}, 0);

console.log(3);

console.log(4);

resolve(true);

});

1. then(v => {

console.log(5);

});

var b = new Promise((resolve, reject) => {

console.log(6);

setTimeout(() => {

Console.log(7);

}, 0);

});

console.log(8)

```

以上代码段使用ES6的新特性，相信你应该有所了解，请给出你的输出结果，请小心输 出先后关系。

5. 有一个数组a[N]顺序存放0~N-1, 要求没隔两个数删掉一个数，到末尾时

循环至开头继续进行，求最后一个被删掉的原始下标位置。以8个数(N=7)为例：

{0, 1, 2, 3, 4, 5, 6, 7}, 0->1->2(删除)->3->4->5(删除)->6->7->0(删除),

如此循环直到最后一个数被删除。算法实现。