变量 a 是一个 64 位有符号的整数，初始值用 16 进制表示为：0Xf000000000000000； 变量 b 是一个 64 位有符号的整数，初始值用 16 进制表示为：0x7FFFFFFFFFFFFFFF。 则 a-b 的结果用 10 进制表示为多少？（）

正确答案: C

```
1
-(2^62+2^61+2^60+1)
2^62+2^61+2^60+1
2^59+(2^55+2^54+…+2^2+2^1+2^0)
```

```
console.log(([])?true:false);
console.log(([]==false?true:false));
console.log(({}==false)?true:false)
```

得到的结果分别是什么？（）

正确答案: D

false true true
true true true
true false true
true true false

下列哪些是块级元素（）

正确答案: B C D E F

```
input
ul
hr
li
div
form
```

关于跨域问题下面说法正确的是？（）

正确答案: B

可以利用 flash 的 http 请求，来处理跨域问题
通过 iframe 设置 document.domain 可以实现跨域
一般情况下，m.toutiao.com 可以 ajax 请求 www.toutiao.com 域名下的接口并获得响应
通过 jsonp 方式可以发出 post 请求其他域名下的接口

以下符合 ES6 写法的有：（）

正确答案: C

A. class Foo

```
{
    constructor() {return Object.create(null);}
}
Foo()
```

B． var m=1;
export m;

C.export var firstName='Michael';

D.在 A 模块中 export{readFile}后，在 B 模块中 import readFile from ‘A’可以获取到 readFile

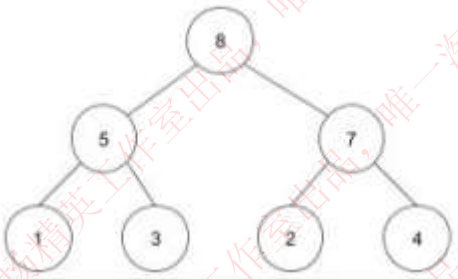
可继承的样式属性包括（）

正确答案: A C

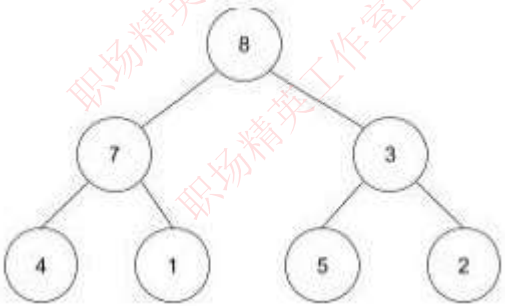
```
color
background-color
font-size
border
margin
```

堆的数据结构能够使得堆顶总是维持最大（对于大根堆）或最小（对于小根堆），给定一个数组，对这个数组进行建堆，则平均复杂度是多少？如果只是用堆的 push 操作，则一个大根堆依次输入 3,7,2,4,1,5,8 后，得到的堆的结构示意图是下述图表中的哪个？（）

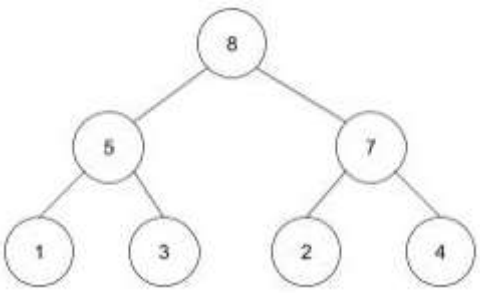
A.O(n)



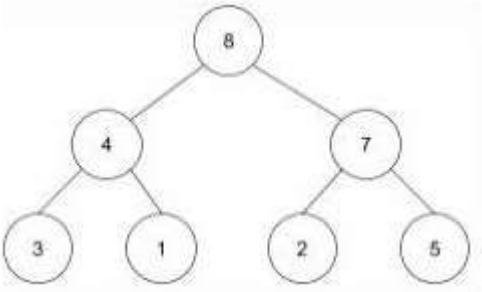
B.O(n) ,



C.O(logn)



D.O(n),



正确答案: D

A
B
C
D

http 请求方式 get 和 post 的区别包括（）

正确答案: A B

get 和 post 的可传输内容大小不一样，一个有限制一个没有限制
get 和 post 传输的内容存放的位置不一样，一个放在 header，一个放在 body
get 请求 Content-type 只能是 text/html
get 请求可以跨域、post 请求不能跨域

下面哪些属于 JavaScript 的 typeof 运算符的可能结果：（）

正确答案: A C F G

symbol
NaN
boolean
null
array
undefined
string

老王有两个孩子，已知至少有一个孩子是在星期二出生的男孩。问：两个孩子都是男孩的概率是多大？

正确答案: A

13/27
7/9
1/2
1/3

下列说法正确的有：（）

正确答案: A B

visibility:hidden;所占据的空间位置仍然存在，仅为视觉上的完全透明；
display:none;不为被隐藏的对象保留其物理空间；
visibility:hidden;与 display:none;两者没有本质上的区别；
visibility:hidden;产生 reflow 和 repaint（回流与重绘）；

TCP 断开连接的四次挥手中，第四次挥手发送的包会包含的标记，最正确的描述是？（）

正确答案: C

FIN
FIN,PSH
ACK
FIN,ACK

页面有一个按钮 button id 为 button1，通过原生的 js 如何禁用？（）

正确答案: B D

```
document.getElementById("button1").setAttribute("Readolny",true);
document.getElementById("button1").setAttribute("disabled","tru
e");
document.getElementById("button1").Readolny=true;
document.getElementById("button1").disabled=true;
```

关于下列 CSS 选择器：ID 选择器、类选择器、伪类选择器、标签名称选择器，排序正确的是：（）

正确答案: D

ID 选择器>Class 选择器>伪类=标签名称选择器
ID 选择器>伪类>Class 选择器>标签名称选择器
ID 选择器>Class 选择器>伪类>标签名称选择器
ID 选择器>Class 选择器=伪类>标签名称选择器

假设 a 是一个由线程 1 和线程 2 共享的初始值为 0 的全局变量，则线程 1 和线程 2 同

时执行下面的代码，最终 a 的结果不可能是（）

```
boolean isOdd = false;

for(int i=1;i<=2;++i)
{
if（i%2==1）isOdd = true；
else isOdd = false；
a+=i*(isOdd?1:-1)；
}
```

正确答案: D

-1
-2
0
1

使用 HTML+CSS 实现如图布局，border-width:5px，格子大小是 50px*50px，hover 时
边框变成红色，需要考虑语义化。


```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
<style type="text/css">
table{
/* border-collapse:separate;*/
border: none;
border-spacing: 0;
}
td{
```

```
position: relative;
width: 50px;
height: 50px;
border: 5px solid blue;
background: #fff;
color: green;
text-align: center;
line-height: 50px;
display: inline-block;
}
tr:not(:first-child) td{
margin-top: -5px;
}
tr td:not(:last-child){
margin-right: -5px;
}
td:hover{
border-color: red;
cursor: pointer;
z-index: 2;
}
</style>
</head>
<body>
<table>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
```

<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</table>
</body>
</html>

z-index 仅能在定位元素上奏效（position 属性值设置除默认值 static 以外的元素，包括 relative，absolute，fixed 样式）

给出一个上传文件时不用刷新页面的方案，要求写出关键部分的 js 代码。

```
<input id="upload" type="file" />
<button id="upload-btn"> upload </button>
```

```
23
24
25
26
27
28
29
30
31
32
33
34
35
document.getElementById('upload-btn').onclick = function(){


    varinput = document.getElementById('upload');

    varfile = input.files[0];

    varformData = newFormData();

    forData.append('file',file);



    fetch({

    url:'/upload',

    mothod:'POST',

    body:formData

    })

    .then((d)=>{

    console.log('result is',d);

    alert('上传完毕');

    })
```

}

给定整数 n 和 m，将 1 到 n 的这 n 个整数按字典序排列之后，求其中的第 m 个数。
对于 n=11, m=4，按字典序排列依次为 1, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9，因此第 4 个数是 2.
对于 n=200, m=25，按字典序排列依次为 1 10 100 101 102 103 104 105 106 107 108 109 11 110 111 112 113 114 115 116 117 118 119 12 120 121 122 123 124 125 126 127 128 129 13 130 131 132 133 134 135 136 137 138 139 14 140 141 142 143 144 145 146 147 148 149 15 150 151 152 153 154 155 156 157 158 159 16 160 161 162 163 164 165 166 167 168 169 17 170 171 172 173 174 175 176 177 178 179 18 180 181 182 183 184 185 186 187 188 189 19 190 191 192 193 194 195 196 197 198 199 2 20 200 21 22 23 24 25 26 27 28 29 3 30 31 32 33 34 35 36 37 38 39 4 40 41 42 43 44 45 46 47 48 49 5 50 51 52 53 54 55 56 57 58 59 6 60 61 62 63 64 65 66 67 68 69 7 70 71 72 73 74 75 76 77 78 79 8 80 81 82 83 84 85 86 87 88 89 9 90 91 92 93 94 95 96 97 98 99 因此第 25 个数是 120...

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (sc.hasNext()) {
            long n = sc.nextLong();
            long m = sc.nextLong();
            System.out.println(solve(n, m));
        }
    }

    private static long solve(long n, long m) {
        long ans = 1;
        while (m != 0) {
            long cnt = getCntOfPre(ans, n);
            if(cnt >= m) {
                m --;
                if(m == 0)
                    break;
                ans *= 10;
            } else {
                m -= cnt;
                ans ++;
            }
        }
        return ans;
```

```
        }

        private static long getCntOfPre(long pre, long n) {
            long cnt = 1;
            long p = 10;
            for (; pre * p <= n; p *= 10) {
                cnt += Math.min(n, pre * p - 1 + p) - pre * p + 1;
            }
            return cnt;
        }
    }
```