[单选题]

1. 攻击者使用无效IP地址，利用TCP连接的三次握手过程，连续发送会话请求，使受害主机处于开放会话的请求之中，直至连接超时，最终因耗尽资源而停止响应。这种攻击被称为

A.[DNS欺骗攻击](javascript:void(0);)

B.[DDoS攻击](javascript:void(0);)

C.[SYN Flooding攻击](javascript:void(0);)

D.[重放攻击](javascript:void(0);)

2. A,B两台机器都正常工作,B机器未监听任何端口.如果A机器向B机器80端口发送SYN包,会收到何种类型的回包?

A.[RST包](javascript:void(0);)

B.[ACK包](javascript:void(0);)

C.[FIN包](javascript:void(0);)

D.[无回应](javascript:void(0);)

3.awk里的内置变量NF指的是

A.[以指定分隔符号作为分隔的列数](javascript:void(0);)

B.AWK处理文件的个数

C.[单个输入文件记录的行数](javascript:void(0);)

D.[所有输入文件记录的行数](javascript:void(0);)

4.堆排序的额外空间复杂度是（）

A.[O(n)](javascript:void(0);)

B.[O(logn)](javascript:void(0);)

C.[O(1)](javascript:void(0);)

D．[视情况而定](javascript:void(0);)

5.给定 `<div id="a" style="float:left;"><div style="width: 100px"></div></div>` `<div id="b"><div style="width: 100px"></div></div>`， `div#a` 和 `div#b` 的宽度分别是【100%即其父容器宽度】

A.[100%, 100%](javascript:void(0);)

B.[100px, 100%](javascript:void(0);)

C.[100%, 100px](javascript:void(0);)

D.[100px, 100px](javascript:void(0);)

6.以下代码的输出结果是

// counter.js

let counter = 10;  
export default counter;  
  
// index.js  
import myCounter from "./counter";  
myCounter += 1;  
console.log(myCounter);

A.[10](javascript:void(0);)

B.[11](javascript:void(0);)

C.[Error](javascript:void(0);)

D.[NaN](javascript:void(0);)

7.对字符串HI\_KWAI中的字符进行二进制编码，使得字符串的编码长度尽可能短，最短长度为

A.[7](javascript:void(0);)

B.[14](javascript:void(0);)

C.[18](javascript:void(0);)

D.[22](javascript:void(0);)

8.无向图G有20条边，有4个度为4的顶点，6个度为3的顶点，其余顶点度小于3，则G最多有多少个顶点

A.[14](javascript:void(0);)

B.[16](javascript:void(0);)

C.[18](javascript:void(0);)

D.[20](javascript:void(0);)

9.3的方幂及不相等的3的方幂的和排列成递增序列{1,3,4,9,10,12,13…}，则数列第100项是

A.[975](javascript:void(0);)

B.[976](javascript:void(0);)

C.[981](javascript:void(0);)

D.[982](javascript:void(0);)

10.最长递增子序列（LIS）问题是指，在一个给定的数值序列中，找到一个子序列，使得这个子序列元素的数值依次递增，并且这个子序列的长度尽可能地大。

比如{2,1,4,2,3,7,4,7,19} 的LIS是 {1,2,3,4,7,19}，长度是6。

考虑大小为N的数组，获取其LIS的最小时间和空间复杂度分别为：

A.[N^2，N^2](javascript:void(0);)

B.[NlogN，N^2](javascript:void(0);)

C.[NlogN，NlogN](javascript:void(0);)

D.[NlogN，N](javascript:void(0);)

11.F(x)=1 (x<=1), F(n)=25\*F(n/5)+n^2 求F(n)的时间复杂度:

A.[O(n\*log(n))](javascript:void(0);)

B.[O(log(n))](javascript:void(0);)

C. O(n)

D.[O(n^2\*log(n))](javascript:void(0);)

12.A，B，C，D都为32位整型，基于以下给定的C，D能否得出A，B

A.[C=A+B,D=A-B](javascript:void(0);)

B.[C=A\*B,D=A/B](javascript:void(0);)

C.[C=A+B,D=B](javascript:void(0);)

D.[C=A-B,D=(A+B)>>1](javascript:void(0);)

13.文法G：S->abSba|c所识别的语言是：

A. (abcba)\*

B.[abc\*ba](javascript:void(0);)

C.[(ab)\*c(ba)\*](javascript:void(0);)

D.[ab(ab)\*c(ba)\*ba](javascript:void(0);)

14.假定+，\*和&分别表示加法、乘法和指数运算，且

计算符优先级顺序从高到低依次为：+，\*和&，

运算符运算时遵循左结合率，则：2+1\*2&1\*2&2的值为：

A.[10](javascript:void(0);)

B.[18](javascript:void(0);)

C.[324](javascript:void(0);)

D.[1296](javascript:void(0);)

15.下面哪个数据库不属于NoSQL的范畴？

A.[HBase](javascript:void(0);)

B.[Cassandra](javascript:void(0);)

C. MongoDB

D.[PostgreSQL](javascript:void(0);)

16.体育课的铃声响了，同学们都陆续地奔向操场，按老师的要求从高到矮站成一排。每个同学按顺序来到操场时，都从排尾走向排头，找到第一个比自己高的同学，并站到他的后面，这种站队的方法类似下列哪种算法（）

A.[冒泡排序](javascript:void(0);)

B.[插入排序](javascript:void(0);)

C.[归并排序](javascript:void(0);)

D.[快速排序](javascript:void(0);)

17.int i=0;

try{

    i++;

    return i;

} catch(Exception e)

{

    i++;

} finally {

    i++;

}

这段代码的返回值是什么

A.[0](javascript:void(0);)

B.[1](javascript:void(0);)

C.[2](javascript:void(0);)

D.[3](javascript:void(0);)

18.当（）时，进程从执行状态转变为就绪状态

A.[进程被调度程序选中](javascript:void(0);)

B.[时间片到](javascript:void(0);)

C.[等待某一事件](javascript:void(0);)

D.[等待的事件发生](javascript:void(0);)

19.在MySQL的众多存储引擎中，只有InnoDB支持事务，关于事物隔离级别，以下说法错误的是（）

A.[Read uncommitted、Read committed 、Repeatable read、Serializabl四种隔离级别并行性能依次降低，安全性依次提高。](javascript:void(0);)

B.[脏读是某一事务读取了另外一个事务未提交的数据，不可重复读是读取了其他事务提交的数据，脏读和不可重复读都可以通过事物隔离级别控制。](javascript:void(0);)

C.[RR隔离级别，只能返回比当前事务早的提交插入、更新、删除值。](javascript:void(0);)

D.[RR和RC隔离级别都存在幻读，RR隔离级别幻读可以通过next-key lock避免。](javascript:void(0);)

20.给定一个字符串集合，集合大小为n，n=6000000, 请问从集合中查询字符串，以下算法中最快速的是哪一个？

A.[skiplist（跳表）](javascript:void(0);)

B.[B 树](javascript:void(0);)

C.[哈希表](javascript:void(0);)

D.[红黑树](javascript:void(0);)

21. [编程题]集合划分问题

时间限制：C/C++ 1秒，其他语言2秒

空间限制：C/C++ 32M，其他语言64M

给定一个数组，每个元素范围是0~K（K < 整数最大值2^32），将该数组分成两部分，使得 |S1- S2|最小，其中S1和S2分别是数组两部分的元素之和。

**输入描述:**

数组元素个数N（N 大于1但不超过 10, 000, 000）  
  
数组中N个元素（用空格分割）

**输出描述:**

|S1- S2|的值

**输入例子1:**

5

2 4 5 6 9

**输出例子1:**

0

**输入例子2:**

4

1 1 1 999

**输出例子2:**

996

22. [编程题]最长等差数列问题

时间限制：C/C++ 1秒，其他语言2秒

空间限制：C/C++ 32M，其他语言64M

给定一个未排序数组,找出其中最长的等差数列(无需保证数字顺序)。

**输入描述:**

第一行N表示数组中元素个数（N < 10,000,000）  
  
第二行是数组的元素，用空格分割

**输出描述:**

等差序列长度

**输入例子1:**

5

1 4 2 5 3

**输出例子1:**

5

23. [编程题]字母组合

时间限制：C/C++ 1秒，其他语言2秒

空间限制：C/C++ 32M，其他语言64M

给定一个仅包含数字 2-9 的字符串，返回所有它能表示的字母组合，按照字典序升序排序,如果有重复的结果需要去重

给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。



**输入描述:**

输入2-9数字组合， 字符串长度 1<=length<=20

**输出描述:**

输出所有组合

**输入例子1:**

23

**输出例子1:**

[ad, ae, af, bd, be, bf, cd, ce, cf]

**输入例子2:**

92

**输出例子2:**

[wa, wb, wc, xa, xb, xc, ya, yb, yc, za, zb, zc]

**输入例子3:**

458

**输出例子3:**

[gjt, gju, gjv, gkt, gku, gkv, glt, glu, glv, hjt, hju, hjv, hkt, hku, hkv, hlt, hlu, hlv, ijt, iju, ijv, ikt, iku, ikv, ilt, ilu, ilv]

24. [编程题]验证IP地址

时间限制：C/C++ 1秒，其他语言2秒

空间限制：C/C++ 32M，其他语言64M

编写一个函数来验证输入的字符串是否是有效的 IPv4 或 IPv6 地址  
  
IPv4 地址由十进制数和点来表示，每个地址包含4个十进制数，其范围为 0 - 255， 用(".")分割。比如，172.16.254.1；  
同时，IPv4 地址内的数不会以 0 开头。比如，地址 172.16.254.01 是不合法的。  
  
IPv6 地址由8组16进制的数字来表示，每组表示 16 比特。这些组数字通过 (":")分割。比如,  2001:0db8:85a3:0000:0000:8a2e:0370:7334 是一个有效的地址。而且，我们可以加入一些以 0 开头的数字，字母可以使用大写，也可以是小写。所以， 2001:db8:85a3:0:0:8A2E:0370:7334 也是一个有效的 IPv6 address地址 (即，忽略 0 开头，忽略大小写)。  
  
然而，我们不能因为某个组的值为 0，而使用一个空的组，以至于出现 (::) 的情况。 比如， 2001:0db8:85a3::8A2E:0370:7334 是无效的 IPv6 地址。  
同时，在 IPv6 地址中，多余的 0 也是不被允许的。比如， 02001:0db8:85a3:0000:0000:8a2e:0370:7334 是无效的。  
  
说明: 你可以认为给定的字符串里没有空格或者其他特殊字符。

**输入描述:**

一个IP地址字符串

**输出描述:**

ip地址的类型，可能为  
  
IPv4, IPv6, Neither

**输入例子1:**

172.16.254.1

**输出例子1:**

IPv4

**例子说明1:**

这是一个有效的 IPv4 地址, 所以返回 "IPv4"

**输入例子2:**

2001:0db8:85a3:0:0:8A2E:0370:7334

**输出例子2:**

IPv6

**例子说明2:**

这是一个有效的 IPv6 地址, 所以返回 "IPv6"

**输入例子3:**

256.256.256.256

**输出例子3:**

Neither

**例子说明3:**

这个地址既不是 IPv4 也不是 IPv6 地址