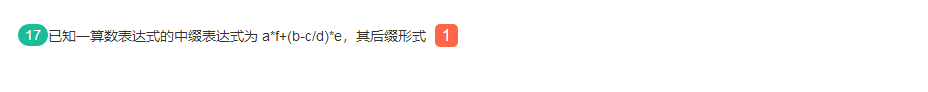
对一棵二叉树进行遍历，我们可以采取3中顺序进行遍历，分别是前序遍历、中序遍历和后序遍历。  
这三种方式是以访问父节点的顺序来进行命名的。假设父节点是N，左节点是L，右节点是R，那么对应的访问遍历顺序如下：  
  
前序遍历    N－>L－>R   ？我们要推断的  
中序遍历    L－>N－>R   MFLEDABKCGHJI  
后序遍历    L－>R－>N   FELMDKHGJICBA   
  
其实，只要知道其中任意两种遍历的顺序，我们就可以推断出剩下的一种遍历方式的顺序，这里我们只是以：知道后序遍历和中序遍历，推断先序遍历作为例子，其他组合方式原理是一样的。要完成这个任务，我们首先要利用以下几个特性：  
特性A，对于前序遍历，第一个肯定是根节点；  
特性B，对于后序遍历，最后一个肯定是根节点；  
特性C，利用前序或后序遍历，确定根节点，在中序遍历中，根节点的两边就可以分出左子树和右子树；  
特性D，对左子树和右子树分别做前面3点的分析和拆分，相当于做递归，我们就可以重建出完整的二叉树；  
  
1.根据特性B，对于后序遍历，最后一个肯定是根节点；得出根节点：A  
根据特性C，在中序遍历中，根节点的两边就可以分出左子树和右子树  
            A  
        /        \  
MFLED   BKCGHJI  
  
2. 取出左子树，在中序的左子树：MFLED    在后序的左子树：FELMD  
根据特性：对于后序遍历，最后一个肯定是根节点；  
得出左子树的父节点是D,并且D没有右子树  
           A  
         /    \  
       D   BKCGHJI  
      /  
 MFLE  
3.使用同样的方法：  
后序是FELM 中序是MFLE  
所以，M为父节点，并且M没有左节点  
                A  
              /    \  
           D   BKCGHJI  
         /  
      M  
        \  
       FLE  
接着后序FEL，中序FLE  
所以L为父节点，F为左节点，E为右节点  
  
                 A  
               /   \  
             D BKCGHJI  
           /   
        M  
           \  
            L  
           /   \  
         F     E  
  
4.取出右子树，中序：BKCGHJI  后序：KHGJICB  
父节点为B，  
            A  
          /    \  
        D      B  
      /             \  
   M           KCGHJI  
     \  
      L  
     /   \  
   F     E  
中序：KCGHJI  后序：KHGJIC,父节点：C  
左节点：K，右节点:GHJI  
  
               A  
             /    \  
           D      B  
         /            \  
      M              C  
       \             /   \  
        L       K    GHJI  
       /  \  
     F    E  
中序：GHJI，后序：HGJI，父节点：I,只有左节点GHJ  
  
                      A  
                    /    \  
                 D       B  
                /           \  
             M            C  
                \          /   \  
                 L      K    I  
                /   \         /  
               F   E    GHJ  
中序：GHJ，后序:HGJ,父节点：J,只有左节点：GH  
                         A  
                       /    \  
                    D      B  
                  /            \  
                M            C  
                  \          /    \  
                   L       K      I  
                 /   \            /  
               F     E      J  
                            /  
                          GH  
中序：GH,后序：HG,父节点：G,只有右节点H  
                                 A  
                               /    \  
                             D      B  
                           /           \  
                         M             C  
                            \           /   \  
                             L       K      I  
                           /   \            /  
                        F      E      J  
                                      /  
                                    G  
                                      \  
                                       H  
进行先序排列：根左右 :A，DMLFE,BCKIJGH  
结果为：ADMLFEBCKIJGH



这里所谓的前缀，中缀，后缀是根据操作符的位置来定的，如果操作符在操作数前面，则称为前缀表达式，例如“- + 1 × + 2 3 4 5”;如果操作符在操作数之间，则称为中缀表达式，例如

“1+((2+3)×4)-5”;如果操作符在操作数后面，则称为后缀表达式，例如“1 2 3 + 4 × + 5 -”。

虽然中缀表达式符合人类的日常思维习惯，但是计算机在存储中缀表达式时，需要使用树这种数据结构，如果表达式过于复杂，那么树的高度会变得很高，大大增加了时间复杂度和空间复杂度。如果转换成线性结构，那么效率将变得高很多，所以需要将中缀表达式先转换成前缀或者后缀表达式，然后依靠栈这种线性数据结构来进行计算。

前缀表达式又叫波兰表达式，后缀表达式又叫逆波兰表达式。前缀表达式基本没有在商业计算机中使用过，所以现实中用的更多的是后缀表达式。

如何将中缀表达式转化成后缀表达式呢？

利用两个栈S1，S2：其中S1存放操作符，S2存放操作数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | 扫描到的元素 | | S2(栈底->栈顶)数字 | S1 (栈底->栈顶)操作符 | 说明 |
| a | a | 空 | 数字，直接入栈 |
| \* | a | \* | S1为空，运算符直接入栈 |
| f | af | \* | 数字，直接入栈 |
| + | af\* | + | +比\*优先级低  将S1中的栈顶元素出栈，并压入S2中 |
| ( | af\* | +( | ( ,直接入栈 |
| b | af\*b | +( | 数字，直接入栈 |
| - | af\*b | +(- | S1栈顶为左括号，运算符直接入栈 |
| c | af\*bc | +(- | 数字，直接入栈 |
| +(-/ | af\*bc |  | /比-优先级高，则直接压栈 |
| d | af\*bcd | +(-/ | 数字，直接入栈 |
| ) | af\*bcd/- | + | 右括号，弹出运算符到s2直至遇到左括号 |
| \* | af\*bcd/- | +\* | \*比+优先级高，则直接压栈 |
| e | af\*bcd/-e | +\* | 数字，直接入栈 |
| 到达最右端 | af\*bcd/-e\*+ | 空 | 弹出S1中剩余的运算符 |

从左往右遍历中缀表达式，如果遇到数字，则放入S2中，如果遇到操作符，则放入S1中。在放操作符的时候有一定的规则，如果栈为空或栈顶元素为（，则直接压栈。如果是(，也直接压栈;如果栈顶元素为普通操作符，则比较优先级，如果待压栈的操作符比栈顶操作符优先级高，则直接压栈，否则将S1中的栈顶元素出栈，并压入S2中，再接着比较S1栈顶元素的优先级。如果遇到)，则依次弹出S1栈顶的运算符，并压入S2，直到遇到左括号为止，此时将这一对括号丢弃。最后将S1中剩余的运算符依次弹出并压入S2，逆序输出S2（从栈底到栈顶）便得到了后缀表达式。（注意：等号的优先级最低，因为要到最后才进行赋值操作）

得到后缀表达式之后，计算就变得方便多了，遇到数字就压栈，遇到操作符的时候，pop出栈顶的两个元素，进行计算后将结果又压入栈中，这样一直下去，直到得到最终结果。

将中缀表达式“a\*f+(b-c/d)\*e”转换为后缀表达式的过程如下：

得出结果：af\*bcd/-e\*+，OK

Ping是Windows、Unix和Linux系统下的一个命令。ping也属于一个通信协议，是TCP/IP协议族的一个子协议。利用"ping"命令可以检查网络是否连通，可以很好地帮助我们分析和判定[网络故障](https://baike.so.com/doc/5709765-5922486.html" \t "_blank)。应用格式:**Ping空格**[IP地址](https://baike.so.com/doc/4252723-4455111.html" \t "_blank)。该命令还可以加许多参数使用，具体是键入Ping按回车即可看到详细说明。使用的是ICMP协议，是“Internet Control Message Protocol”（Internet控制消息协议）的缩写用于在IP[主机](http://www.so.com/s?q=%E4%B8%BB%E6%9C%BA&ie=utf-8&src=internal_wenda_recommend_textn)、[路由器](http://www.so.com/s?q=%E8%B7%AF%E7%94%B1%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_textn)之间传递控制消息。

【测试方向优先】你用浏览器打开一个网站，却没有按预期看到应有的网页内容。请分析各种可能的原因，如果这些原因表现出来的现象不同，也请描述

参考要点：

人的原因：弄错网址；拼写错误；未打开网络；。。。

本机原因：浏览器版本低不兼容网页；本机防火墙阻止访问；中病毒导致浏览器工作不正常；。。。

服务端原因：服务器宕机；服务器出错；服务器高负载无法及时回应；服务器超负载主动拒绝回应；本机被服务端加入了黑名单；

网络原因：网络不通；网络拥塞；DNS解析失败；DNS解析到错误的IP；。。。

其他环境原因：域名被劫持；访问被黑客攻击；。。。

【运维方向优先】你用浏览器打开一个电商网站，准备浏览购物。从你输入完网站的网址敲下Enter键，到网站首页迅速呈现出来的这段时间，你的电脑系统里发生了哪些事情？请根据你所学的计算机知识，尽可能详细的解释。（提示：从软硬件的尽量多的层次考虑和描述。）

参考要点：回答是否从硬件（键盘网卡）、OS、TCP协议栈、运行库、浏览器、HTTP、HTML/JS等多层面描述。

参考回答：键盘硬件中断；OS处理中断，转换为特定消息放入浏览器程序的事件队列；浏览器的消息循环处理该消息，请求网址；OS请求本地域名缓存或域名服务器解析网址中的域名，得到IP；浏览器向该IP建立TCP连接（默认80端口）；浏览器发送GET请求，包含网站的路径，TCP协议栈组装为TCP包，通过网卡发送；浏览器等待网站回复，进程被OS切换为等待状态；网站返回的数据到来，网卡产生中断；OS处理中断，TCP协议栈将数据读入buffer；浏览器获得数据，处理HTTP头，显示HTML网页

*更多*：OS发ARP包获得网关MAC地址，所有DNS请求、网站TCP等数据包均发向该网关；浏览器IO等待期间OS切换运行系统中其他进程；浏览器将HTTP头中解析出的cookie保存到文件系统；根据网页内容发起更多的HTTP请求获取图片、运行内嵌的javascript脚本等；将网页按照HTTP 头的指示缓存；将网址加入浏览历史保存到文件系统；浏览器整个处理过程中，运行库和OS对内存做相应分配释放，磁盘做相应的IO操作；。。。

答:1）交换机  
在计算机网络系统中，交换机是针对共享工作模式的弱点而推出的。交换机拥有一条高带宽的背部总线和内部交换矩阵。交换机的所有的端口都挂接在这条背 部总线上，当控制电路收到数据包以后，处理端口会查找内存中的地址对照表以确定目的MAC（网卡的硬件地址）的NIC（网卡）挂接在哪个端口上，通过内部 交换矩阵迅速将数据包传送到目的端口。目的MAC若不存在，交换机才广播到所有的端口，接收端口回应后交换机会“学习”新的地址，并把它添加入内部地址表 中。  
交换机工作于OSI参考模型的第二层，即数据链路层。交换机内部的CPU会在每个端口成功连接时，通过ARP协议学习它的MAC地址，保存成一张 ARP表。在今后的通讯中，发往该MAC地址的数据包将仅送往其对应的端口，而不是所有的端口。因此，交换机可用于划分数据链路层广播，即冲突域；但它不 能划分网络层广播，即广播域。  
交换机被广泛应用于二层网络交换，俗称“二层交换机”。  
交换机的种类有：二层交换机、三层交换机、四层交换机、七层交换机分别工作在OSI七层模型中的第二层、第三层、第四层盒第七层，并因此而得名。  
2）路由器  
路由器（Router）是一种计算机网络设备，提供了路由与转送两种重要机制，可以决定数据包从来源端到目的端所经过 的路由路径（host到host之间的传输路径），这个过程称为路由；将路由器输入端的数据包移送至适当的路由器输出端(在路由器内部进行)，这称为转 送。路由工作在OSI模型的第三层——即网络层，例如网际协议。  
路由器的一个作用是连通不同的网络，另一个作用是选择信息传送的线路。 路由器与交换器的差别，路由器是属于OSI第三层的产品，交换器是OSI第二层的产品(这里特指二层交换机)。  
3）网关  
网关（Gateway），网关顾名思义就是连接两个网络的设备，区别于路由器（由于历史的原因，许多有关TCP/IP 的文献曾经把网络层使用的路由器（Router）称为网关，在今天很多局域网采用都是路由来接入网络，因此现在通常指的网关就是路由器的IP），经常在家 庭中或者小型企业网络中使用，用于连接局域网和Internet。 网关也经常指把一种协议转成另一种协议的设备，比如语音网关。  
在传统TCP/IP术语中，网络设备只分成两种，一种为网关（gateway），另一种为主机（host）。网关能在网络间转递数据包，但主机不能 转送数据包。在主机（又称终端系统，end system）中，数据包需经过TCP/IP四层协议处理，但是在网关（又称中介系 统，intermediate system）只需要到达网际层（Internet layer），决定路径之后就可以转送。在当时，网关 （gateway）与路由器（router）还没有区别。  
在现代网络术语中，网关（gateway）与路由器（router）的定义不同。网关（gateway）能在不同协议间移动数据，而路由器（router）是在不同网络间移动数据，相当于传统所说的IP网关（IP gateway）。  
网关是连接两个网络的设备，对于语音网关来说，他可以连接PSTN网络和以太网，这就相当于VOIP，把不同电话中的模拟信号通过网关而转换成数字信号，而且加入协议再去传输。在到了接收端的时候再通过网关还原成模拟的电话信号，最后才能在电话机上听到。  
对于以太网中的网关只能转发三层以上数据包，这一点和路由是一样的。而不同的是网关中并没有路由表，他只能按照预先设定的不同网段来进行转发。网关最重要的一点就是端口映射，子网内用户在外网看来只是外网的IP地址对应着不同的端口，这样看来就会保护子网内的用户。

**以windows对文件的复制粘帖功能为例，尽可能多地写出测试思路。**

抓住关键字，然后进行发散。

1. 基本功能测试： 文件的复制粘贴功能，首先关键字“文件”，文件有不同的分类（图片、视频、音频、文档等），每个分类又有不同的类型（文档类型：txt doc execl pdf等），每个文件又有不同的大小，而且文件还有很多权限，是不是隐藏，是不是只是管理员可执行。选择不同分类的不同类型，不同大小的文件做测试资源。比如：文档类型里面txt文件可以分为 1.KB的txt文件、1MB的txt文件、1GB的txt文件。。。。下一个关键字 复制粘贴 复制有多种方式 右击选择、Ctrl+C、 拖动复制，对应粘贴也有各种方式。然后从哪复制，粘贴到哪，比如 可以有本机硬盘、移动硬盘、优盘、内存卡、软盘、光盘、连接手机存储，复制到网络地址等等。复制粘贴后文件是不是可用，文件权限是不是有变化。复制过去容量不够怎么处理？复制过后有重名文件怎么处理？复制过程中取消、关机、拔优盘怎么处理？复制过程能不能执行文件？

2.性能测试：复制粘贴功能性能怎么样？复制文件的速度可不可以接受？同时复制多个文件是不是可以完成?复制文件过程中占用CPU资源大不大，耗电量大不大？

3.兼容性测试 Windows XP, Windows 7, Windows 8 , Windows 8.1, Windows 10等各种windos版本是不是都支持这个功能。

4.交互测试; 复制粘贴文件时，使用windows存储的其他功能是否有影响？比如播放本地的音频、视频、等同时复制文件是不是有影响。一边复制，一边粘贴是不是有影响。

**已知String convert(String page)作用是将WEB页转码为方便移动设备查看的页面，为了确保转码的正确性，请设计相应测试策略。**

1. 使用不同的移动设备查看转换后的页面，检查内容是否正确；
2. 输入正确的网址，进行转码，检查内容是否正确；
3. 输入错误的网址，进行转码，系统是否有相应的提示；
4. 测试转码的速度，或者系统的相应时间；
5. 测试转码后，页面显示是否美观；
6. 输入的web的page为空，是否抛出异常；

【基本功能测试】

功能：

1. 输入正确的网址，进行转码，检查内容是否正确；

边界：

1. 输入信息量非常大的网址
2. 输入没有信息的网址

负值：

1. 输入错误的网址，进行转码，系统是否有相应的提示；
2. 输入的 web 的 page 为空，是否抛出异常；

【兼容测试】

1. 使用不同操作系统的移动设备查看转换后的页面，检查内容是否正确（配置测试；
2. 使用不同分辨率的移动设备查看转换后的页面，检查内容是否正确；
3. 使用不同浏览器查看转换后的页面，检查内容是否正确；

【性能测试】

1. 测试转码的速度，或者系统的响应时间；

【易用性 / 用户体验测试】

1. 测试转码后，页面显示是否美观；