* 【笔试部分】
* 【单选】

1.对于堆和栈的描述，正确的是（）

A.栈的大小会随着函数调用层次的变化而变化

B.从堆上申请内存，没有释放的话，会造成内存泄漏；栈的内存由编译器管理，不会内存泄漏

C.在windows上，堆的生长方向是向下的，随着内存地址减小的方向增长；栈的生长方向是向上的，即向着内存地址增加的方向增长

D.在C语言中，指针只能指向堆上的内存，不能指向栈上的内存

【考查知识点】堆与栈

【解析】栈一般都一个默认大小，栈的地址空间是连续的，如果任其增长会给内存带来困难，A错误；栈由编译器在需要的时候分配，不需要的时候自动清除，而堆是由程序员来释放的，容易产生内存泄漏，而栈是由系统管理的，B正确。对于堆来说，生长方向是向上的，也就是向着内存地址增加的地方；对于栈来说，生长方向是向下的，是向着内存地址减小的方向增长，如下图所示，C错误；C语言中指针也可以指向栈，如char buf[10]={0};char \*buff=buf;buff[1]=’a’;D错误。

|  |
| --- |
| 低　　　　->｜－－－－－－－－－－－－－－－－－｜  ｜　全局量（所有已初始化量 .data，　 ｜  ｜　未初始化量 .bss ）　　　　　　　 ｜  堆起始->｜－－－－－－－－－－－－－－－－－｜  ｜　　　　堆向高地址增长　　　　　　｜  ｜　　　　　　　　　　　　　　　　　｜  ｜　　　　　　　　　　　　　　　　　｜  ｜　　　　　自由空间　　　　　　　　｜  ｜　　　　　　　　　　　　　　　　　｜  ｜　　　　　　　　　　　　　　　　　｜  ｜　　　　栈向低地址增长　　　　　　｜  高　栈起始->｜－－－－－－－－－－－－－－－－－｜ |

2.对于线程和进程的描述，正确的是

A.进程退出前，需要手动销毁所有线程，否则进程退出后，依然会有线程在运行，可能导致不可预测的后果

B.在单核的CPU上使用多线程，并不会比单线程有优势

C.进程是调度和拥有资源的基本单位

D.使用全局变量、内存时，需要使用同步机制，因为它们在同一地址空间内

【考查知识点】线程与进程

【解析】通常一个任务不光CPU要花时间，I/O也要花时间，一个进程等I/O时，CPU是闲置的，另一个进程可以利用CPU进行计算，单核多线程指的是单核CPU轮流执行多个线程，通过给每个线程分配CPU时间片来实现，因为时间片非常短用户感觉是多个线程同时执行，B错误；进程是拥有资源的基本单位，线程是独立调度的基本单位，C错误；D正确。

3.某二叉树有2000个节点，则二叉树的最小高度为：

A.9  B.10    C.11    D.12

【考查知识点】二叉树

【解析】最低为log*2(n+1)*https://uploadfiles.nowcoder.com/images/20190919/56_1568900435177_29C080A5413E925FE3B3CCB4048AB99B ，最高为n，故答案选C

4.在C++中，定义“int \*p=new int(10)”，释放p指向的内存，语句是（）

A.delete \*p         B.delete &p        C.delete p       D.delete []p

【考查知识点】指针

【解析】int \*p=new int-分配一个int型变量所占大小的空间，并将首地址赋给int\*型指针p；int \*p=new int[10]-分配一个有10个int型元素的数组所占空间，并将该数组的第一个元素的地址赋给int\*型指针p；int \*p=new int(10)-分配一个int型变量所占大小的空间，并赋值10。一般用法是new一个数组的话用delete []，其他直接用delete即可。

5.某函数有且只有2个输入参数，参数A的取值可划分为5个等价类，参数B的取值可划分为3个等价类。至少应为函数设计（）组测试数据

A.4      B.10          C.15          D.30

【补充】等价类划分是一种典型的黑盒测试方法。在测试时分为一般等价类测试、健壮等价类测试。

一般等价类测试中包括弱一般等价类（“弱”指含单缺陷设计）和强一般等价类（“强”指含多缺陷设计），“一般”指不考虑无效值。弱一般等价类测试用例通过使用一个测试用例中的每个有效等价类（区间）的代表值来实现（常以对称方式来标识这些测试用例，且注意单边假设作用）；强一般等价类测试用例通过每个独立变量的有效等价类的笛卡尔积来实现。

弱健壮等价类测试。弱健壮等价类测试中的无效测试用例只含一个无效值，其他都是有效值，即含有单缺陷假设。【注：“弱”是指含单缺陷假设，“强”是指含多缺陷假设；“健壮”是指考虑无效值】； 强健壮等价类测试考虑了更多的无效值情况。强健壮等价类测试中的无效测试用例可以包含多个无效值，即含有多个缺陷假设。NextDate函数有三个变量，故相应地强健壮等价类测试用例可包含一个无效值、两个无效值或三个无效值。【注：“强”是指含多缺陷假设，“强”是指含多缺陷假设；“健壮”是指考虑无效值】

【解析】（求大神解答）

6.下列哪些linux的跨进程通信方式不适合数据传输（）

A.pipe，管道   B.shared memory，共享内存   C.socket，套接字     D.semaphore，信号量

【考察知识点】进程通信

【解析】linux进程通信的6种方式有管道、信号量、信号、消息队列、共享内存和套接字。**管道**允许一个进程和另一个与它有共同祖先的进程之间通信；**信号量**作为进程间及同进程不同线程间的同步手段；信号可用于进程间通信和进程本身通信；消息队列是消息的链表，允许一个或多个进程向它写入与读取消息；**共享内存**使多个进程访问同一块内存空间；**套接字**用于不同机器之间的进程间通信。选A

7.如果计算机内存大小为1G，虚拟地址空间为4G，在计算机系统上，需要的指针长度为（）位

A.32          B.30        C.64         D.16

【考查知识点】虚拟内存

【解析】最大寻址空间为4G，则指针为32位，232=4GB

8.在32位系统上，下面程序的输出结果是（）

#include<stdio.h>

void Foo(char param[20]) {

printf("%d\n", sizeof(param));

printf("%d\n", strlen(param));

}

int main(int argc, const char \* argv[]) {

char str1[] = "Hello\0dji";

char str2[] = { 'H','e','l','l','o','\0','d','j','i' };

printf("%d\n", sizeof(str1));

printf("%d\n", sizeof(str2));

Foo(str1);

}

A.10 9 4 5          B.10 9 20 5          C.9 9 20 5        D.9 9 9 5

【考查知识点】sizeof

【解析】因为str1‘\’和‘0’是分开计算而str2是合起来计算的，所以sizeof(str1)为10，sizeof(str2)为9；Foo函数中，数组作为参数传递时会退化成指针，在32位操作系统中，指针占4个字节（64位操作系统占8个字节）；strlen是计算char的长度，以’\0’为字符串结束的标志，所以传入Hello，为5.

9.使用以下哪种技术无法伪装自己的IP地址（）

A.HTTP Proxy          B.Socks Prox       C.VPN           D.修改DNS

【解析】HTTP Proxy是HTTP代理，其中的一个作用就是访问，通过删除HTTP报文中的身份特性从而对远端服务器隐藏原始用户的IP地址及其他细节，同时HTTP代理也不会记录用户访问记录的log，A正确；Socks Prox是Socket代理，可以将浏览器的上网行为保护起来，如果需要所有互联网访问保护起来就需要使用VPN，因此BC正确；修改DNS可以提高上网的速度，可以访问某些因为域名解析存在问题而不能访问的网站，可以屏蔽运营商的广告避免被钓鱼的危险，但并不能伪装IP地址，D错误。

* 【双选】

1.有一项考试的成绩，目前是按照完成时间进行升序排列的，需要按照考试成绩进行排序，成绩高的排在前面，如果成绩相同，则完成时间短的排在前面。以上场景适合采用下列哪些排序算法？

A.插入排序   B.快速排序     C.堆排序    D.冒泡排序

【解析】本题实际考察哪种排序算法是稳定的，插入、冒泡是稳定的，快速和堆是不稳定的。

2.下列关于HTTP协议，描述正确的是

A.HTTP协议工作在应用层

B.HTTP只能使用80端口，HTTPS只能使用443端口

C.HTTP是有状态协议

D.HTTP协议支持一定时间内的TCP连接保持，这个连接可以用于发送/接收多次请求

【解析】HTTP是应用层协议，A正确；HTTP常用端口80/8080/3128/8081/9098，HTTPS默认端口TCP/443，UDP/443，B错误；HTTP是无状态协议，协议的状态是指下一次连接而维护这次连接所传输的信息，比如客户获得一张网页之后关闭浏览器，然后再一次启动浏览器，再登陆该网站，但是服务器并不知道客户关闭了一次浏览器。由于Web服务器要面对很多浏览器的并发访问，为了提高Web服务器对并发访问的处理能力，在设计HTTP协议时规定Web服务器发送HTTP应答报文和文档时，不保存发出请求的Web浏览器进程的任何状态信息。这有可能出现一个浏览器在短短几秒之内两次访问同一对象时，服务器进程不会因为已经给它发过应答报文而不接受第二期服务请求。由于Web服务器不保存发送请求的Web浏览器进程的任何信息，因此HTTP协议属于无状态协议，可以通过Cookie和Session将状态分别保存在客户端和服务器端，C错误；HTTP是建立在TCP协议之上的，建立和释放连接的时间都很短，所以HTTP是一种短连接，为了解决每次连接释放效率低的问题，提出keep-alive保持连接特性，可以在一段时间内多次发送、接收，D正确。

* 【编程题】

给出一串阿拉伯数字，输出对应的中文读法。

输入描述：

输入m，范围是0<=m<100000000

样例：

18

输出：十八

#include<iostream>

using namespace std;

void fun(long n)

{

if (n == 0)

{

cout << 0;

return;

}

int temp = 0;

long arr[5] = { 0 };

int i = 0;

int j = 4;

while (n)

{

temp = n % 10;

arr[j] = temp;

n /= 10;

--j;

++i;

}

int k = 0;

while (i)

{

k = 5 - i;

switch (i)

{

case 1:if (arr[k] != 0)

{

cout << arr[k];

}

break;

case 2:if (arr[k] != 0)

{

cout << arr[k] << "S";

}

else if (arr[k] == 0 && arr[k+1] != 0 )

{

cout << "L";

}

break;

case 3:if (arr[k] != 0)

{

cout << arr[k] << "B";

}

else if (arr[k] == 0 && arr[k + 1] != 0 && (arr[3] ||arr[4]))

{

cout << "L";

}

break;

case 4:if (arr[k] != 0)

{

cout << arr[k] << "Q";

}

else if (arr[k] == 0 && arr[k + 1] != 0 && (arr[2] ||arr[3]))

{

cout << "L";

}

break;

case 5:if (arr[k] != 0)

{

cout << arr[k] << "W";

}

break;

default:

break;

}

--i;

}

}

int main()

{

/\*

12001 1W2QL1

\*/

long n = 0;

while (1)

{

cin >> n;

if (n >= 0 && n <= 99999)

{

fun(n);

cout << endl;

}

}

return 0;

}