单选题

1、

int foo(int n){

  if(n<=1)

  return 1;

return n\*foo(n-1);

}

上面算法的时间复杂度是（   ）

A.  O(n2)

B.  O(log2n)

C.  O(n)

D.  O(nlog2n)

参考答案：C

解析：

当n<=1时执行return 1这一个语句。

每次返回上一层都执行n\*foo(n-1)这一个语句，共执行n-1次。

因此共执行基本语句n次，时间复杂度为O(n)。

2、

 public class Test{

  public static void main(String[] args){

   system.out.println(“return value of getValue():” + getValue();

}

  public static int getValue(){

int i = 1;

try{

  i = 4;

}

finally{

  i++;

return i;

     }

}

}

A.  return value of getValue(): 1

B.  其他选项都不对

C.  return value of getValue(): 4

D.  return value of getValue(): 5

参考答案：D

解析：

finally块会执行，执行之前i的值在try块中被修改为5，在finally块内经历++操作之后值为5，最后返回。因此，输出结果为“return value of getValue(): 5”。答案为D。

3、以下程序的输出是（   ）

public class ClassA

{

  private String ClassAName = “ClassA”;

  public ClassA()

{

   print();

}

public void print()

{

   system.out.println(ClassAName);

}

static class ClassB extends ClassA

{

private String ClassAName = “ClassB”;

public void print()

{

 system.out.println(ClassAName);

}

}

public static void main(String[] args)

{

classA b = new ClassB();

}

}

A.  ClassB

B.  Class A ClassB

C.  null

D.  ClassA

参考答案：C

程序代码有部分错误，正确代码：

static public class ClassA {

                  private String ClassAName = "ClassA";

                  public ClassA() {

                           print();

                  }

                  public void print() {

                           System.out.println(ClassAName);

                  }

         }

         static class ClassB extends ClassA {

                  private String ClassAName = "ClassB";

                  public void print() {

                           System.out.println(ClassAName);

                  }

         }

         public static void main(String[] args) {

                  ClassA b = new ClassB();

         }

4、有函数int func(int i)的实现为

 int func(int i)

{

 if(i>0)

  return i\*func(i-2);

else

 return 1;

}

函数调用f(7)的返回值是（   ）

A.  42

B.  35

C.  105

D.  95

参考答案：C

5、表达式(דA˅B)^(C˅D)的逆波兰式表示为（  ）

A.  דAB˅^CD˅

B.  AדB˅CD˅^

C.  AB˅דCD˅^

D.  AדB˅^CD˅

参考答案： B

6、以下二维数组声明合法的是（   ）

A.  char[][] ch = new char[2][3]

B.  char[2][3] ch = new char[][]

C.  char[2][3] ch = new char[][]

D.  char[][] ch = new [2]char[3]

参考答案：A

7、以下代码执行后输出是结果为（   ）

public class Test

{

  public static Test t1 = new Test();

    {

System.out.println(“blockA”);

    }

    static

    {

  System.out.println(“blockB”);

    }

    public static void main(String[] args)

    {

Test t2 = new Test();

     }

}

A.  blockAblockAblockB

B.  blockBblockBblockA

C.  blockAblockBblockA

D.  blockBblockAblockB

参考答案：C

8、以下代码执行后输出结果为（   ）

public class Test{

  public static void main(String[] args){

   System.out.println(“return value of getValue():” +getValue();

}

 Public static int getValue(){

   try{

     return 0;

   } finally{

     return 1;

     }

   }

 }

A.  return value of getValue(): 1

B.  return value of getValue(): 0

C.  return value of getValue(): 1

return value of getValue(): 0

D.  return value of getValue(): 0

return value of getValue(): 1

参考答案：A

9、以下代码执行后输出结果为（   ）

public class Test{

     String str = new String(“hello”);

     char[] ch = {‘a’, ’b’, ’c’};

     public void fun(String str,char ch[]){

        str=”world”;

        ch[0]=’d’;

      }

public static void main(String[] args) {

   ClassTest test1 = new ClassTest();

   test1.fun(test1.str,test1.ch);

   System.out.println(test1.str + “and”);

   System.out.println(test1.ch);

  }

}

A.  hello and abc

B.  world and dbc

C.  hello and dbc

D.  world and abc

参考答案：C

运行结果为：

helloand

dbc

10、通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应的位置并插入的排序算法是（   ）

A.  归并排序

B.  希尔排序

C.  选择排序

D.  插入排序

参考答案：D

11、有正则表达式”\d{3,4}[-]?\d{6,8}”，可能代表的意思是？（   ）   B

A.  手机号码

B.  电话号码

C.  一组数字

D.  QQ号码

12、

public class Test {

  static int cnt = 6;

  static {

     cnt +=9;

}

public static void main(String[] args) {

 System.out.println(“cnt=”+cnt);

}

 static{

  cnt/=3;

};

}

A.  cnt=2

B.  cnt=3

C.  cnt=5

D.  cnt=6

参考答案：C

13、计算具有14个关键字的有序表，折半查找的平均查找长度（   ）

A.  3

B.  20/7

C.  45/14

D.  9

参考答案：C

解析：先花一个二叉查找树，然后根据如下公式计算：

1/14\*(1\*1+2\*2+3\*4+4\*7)=45/14

14、设某操作系统中有5个进程，它们的到达时间和服务时间如下：

  进程     就绪时间     执行时间

   1          0            3

   2          2            6

   3          4            4

   4          6            5

   5          8            2

若采用高响应比优先调度算法，忽略I/O及其他开销时间，其平均周转时间为（   ）

A.  8

B.  7

C.  4

D.  5

参考答案：A

15、一棵完全二叉树有600个节点，那么它的叶子节点有（   ）个

A.  301

B.  300

C.  201

D.  200

参考答案：B

解析：

（600+1）/2=300

16、有一个100\*90的稀疏矩阵，非0元素有20个，设每个整型数占2字节，则用三元组表示该矩阵时，所需的字节数是（   ）

A.  126

B.  120

C.  200

D.  66

参考答案：D

每个元素要用行号,列号,元素值来表示,在用三元组表示稀疏矩阵,还要三个成员来记住,矩阵的行数列数,总的元素数,所以所需的字节数是10\*（1+1+1）\*2+3\*2=66

17、实现线程同步不可以使用下列哪些方法（   ）

A.  管道

B.  互斥量

C.  临界区

D.  信号量

参考答案：A

18、现有教师关系Teacher（教师编号，姓名，年龄，性别，家庭住址），现在要查询姓“李”的且家庭住址包含“西安市”的教师，则筛选条件是（    ）

A.  姓名LIKE’李%’&&家庭住址LIKE’%西安市%’

B.  姓名LIKE’李%’And家庭住址LIKE’%西安市%’

C.  姓名=’李%’And家庭住址=’%西安市%’

D.  姓名=’李%’&&家庭住址=’%西安市%’

参考答案：B

解析：用like进行模糊匹配，且关键字为“AND”。

19、在含有50个结点的二叉排序树上，查找关键字为20的结点，则依次比较的关键字有可能是（  ）

A.  15,35,18,14,20

B.  35,25,18,15,20

C.  15,35,25,20

D.  35,25,28,15,20

参考答案：C

20、若栈S的初始状态为空，元素a,b,c,d,e依次通过栈S，若出栈顺序为c,e,d,b,a，则栈S的容量至少应该为（   ）

A.  5

B.  6

C.  4

D.  3

参考答案：C

备注：出栈顺序题目描述有误，应该为“c,e,d,b,a”。

多选题

1、关于顺序存储结构说法正确的是（   ）

A.  对任何数据结构链式存储结构一定优于顺序存储结构

B.  在顺序存储结构中，执行插入、删除运算会引起相应结点的大量移动

C.  在顺序存储结构中，有时也存储数据结构中元素之间的关系

D.  在顺序存储结构中存储空间已满继续插入新元素时，就会发生“上溢”错误

参考答案：B、D

2、缺页处理过程中，操作系统执行的操作可能是（  ）

A.  修改页表

B.  内存校验

C.  分配页框

D.  磁盘I/O

参考答案：A、C、D

3、对下列常见的各种网络术语描述正确的是（   ）

A.  Telnet是标准的提供远程登录功能的应用，可以在不同OS系统的主机之间运行

B.  Ping是对两个TCP/IP系统连通性进行测试的基本工具，它利用ICMP进行基本的请求和应答

C.  ADNS是一种用于TCP/IP应用程序的分布式数据库，因此它在TCP/IP体系结构中处于应用层

D.  TFIP是一种文件传递应用程序，它使用的传输层协议是TCP

参考答案：A、B

解析：

C、ADNS是硬件防火墙的意思，如果由于书写错误，是DNS的话，则C选项正确。

D、D选项书写有误，没有“TFIP”，应该是“TFTP”。TFTP（Trivial File Transfer Protocol,简单文件传输协议）是TCP/IP协议族中的一个用来在客户机与服务器之间进行简单文件传输的协议，提供不复杂、开销不大的文件传输服务。基于UDP协议实现，端口号为69。

4、有关Java volatile关键字的含义说法正确的是（   ）

A.  线程在每次使用volatile变量的时候，都会获取变量修改后的值

B.  对volatile变量进行“++”读写操作时会被当做原子操作执行

C.  Volatile变量对所有线程是立即可见的

D.  数组元素不能被声明为volatile

参考答案：C、A

解析：

所谓 volatile的措施，就是

1. 每次从内存中取值，不从缓存中什么的拿值。这就保证了用 volatile修饰的共享变量，每次的更新对于其他线程都是可见的。

2. volatile保证了其他线程的立即可见性，就没有保证原子性。

3.由于有些时候对 volatile的操作，不会被保存，说明不会造成阻塞。不可用与多线程环境下的计数器。

5、下面问题能使用贪心法解决的是（   ）

A.  N皇后问题

B.  单源最短路径问题

C.  最小花费生成问题

D.  背包问题

参考答案：B、D、C

解析：

A、N皇后问题是回溯法应用的经典案例。

C、最小花费生成问题是动态规划的经典案例。

6、实现线程同步可以使用下列哪些方法（   ）

A.   管道

B.   互斥量

C.   临界区

D.   信号量

参考答案：B、C、D

7、在Java中重写方法应遵循规则的包括（   ）

A.  参数列表必须完全与被重写的方法相同

B.  必须具有不同的参数列表

C.  可以有不同的访问修饰符

D.  访问修饰符的限制一定要大于被重写方法的访问修饰符

参考答案：A、C

解析：

重写是子类对父类的允许的访问的方法的实现进行重新编写，返回值和形参都不能改变，即外壳不变，核心重写，重写的好处在于子类可以根据需要，定义属于自己的特定行为，也就是子类可以根据需要实现父类的方法，重写方法不能抛出新的检查异常，或者比被重写方法声明更广泛的异常。

         总的原则如下：

方法名相同，参数类型相同，

子类返回的类型等于父类返回的类型

子类抛出的异常小于等于父类抛出的异常，

子类访问权限大于等于父类的访问权限

声明为final的方法不能被重写，

声明为static的方法不能被重写，

8、以下集合对象中哪几个是线程安全的（   ）

A.  ArrayList

B.  Hashtable

C.  LinkedList

D.  Vector

参考答案：B、D

9、Factory Method模式和Prototype模式之间的区别是（   ）

A.  Prototype模式是利用现有的对象进行克隆

B.  Factory Method模式是重新创建一个对象

C.  Prototype模式是重新创建一个对象

D.  Factory Method模式是利用现有的对象进行克隆

参考答案：A、B

解析：

原型模式虽然是创建型的模式，但是与工厂模式没有关系，从名字即可看出，该模式的思想就是将一个对象作为原型，对其进行复制、克隆，产生一个和原对象类似的新对象。

10、在数据库中，事务是并发控制的基本单位。如果对数据库的并发事务不进行控制，则容易发生（   ）问题

A.  丢失修改

B.  不可重复读

C.  读“脏”数据

D.  数据库文件损坏

参考答案：A、B、C

解析：

不对事务进行并发控制，可能会造成前一个进程修改的数据，被后一个进程覆盖写的情况，从而造成丢失修改；多个读进程在读的过程中，中间发生了写，这样前面的读的数据的一样的，后面的读的数据就是不一样的，从而造成不可重复读的问题；一个进程读取了前一个进程还未提交的中间结果，造成读取到脏数据的问题。

11、关于TCP协议描述正确的是（   ）

A.  提供可靠交付的服务

B.  通过窗口大小进行流量的控制

C.  基于字符流的传输层通信协议

D.  提供半双工通信

参考答案：A、B

解析：

C：TCP是字节流的传输层通信协议

D：TCP提供全双工通信

编程题：

数列

**时间限制：**C/C++语言 1000MS；其他语言 3000MS  
**内存限制：**C/C++语言 65536KB；其他语言 589824KB

**题目描述：**

某种特殊的数列a1, a2, a3, ...的定义如下：a1 = 1, a2 = 2, ... , an = 2 \* an − 1 + an - 2 (n > 2)。

给出任意一个正整数k，求该数列的第k项模以32767的结果是多少？

输入

第1行是测试数据的组数n，后面跟着n行输入。每组测试数据占1行，包括一个正整数k (1 ≤ k < 1000000)。

输出

n行，每行输出对应一个输入。输出应是一个非负整数。

样例输入

2

1

8

样例输出

1

408

合法用户名

**时间限制：**C/C++语言 3000MS；其他语言 5000MS  
**内存限制：**C/C++语言 65536KB；其他语言 589824KB

**题目描述：**

WEB应用越来越普及，因此在WEB应用系统中找到一个合意的用户名越来越难。这不，小B所在的公司先前发布了一款火爆的网游，取得了极大的成功，因此公司又在着手开发游戏的续集。为尽快上市，公司安排小B开发用户注册系统。由于英雄所见略同的原因，一个威武霸气的用户名对游戏玩家而言可不是那么容易抢到的。

考虑用户注册的喜好因素，并为用户提供尽可能多的帮助，小B设计了一个用户名注册策略。接收到用户的注册申请时，小B先检查用户数据库，查看申请的用户名是否已经存在。若不存在，则接受该注册申请并将用户名插入到数据库中。若申请的用户名已经存在，则按照一定的规则生成一个新的用户名反馈给用户作为提示，同时插入数据库中。新用户名的生成规则为：从1开始依次向申请的用户名后面增加数字编号，如name1, name2, … , nameN，直至找到一个最小的编号使得新名字与数据库中的用户名不冲突。

小B觉得这个问题不太难，她想让你来解决这个问题，顺便考察一下你的编程水平，你愿意吗？

输入

输入有若干组，每组的第一行为一个整数n，1<=n<=10^5。随后的n行为各个用户申请的用户名，每个用户名为一个由不超过32个小写拉丁字母构成的非空字符串。

输出

对每组中的每个用户注册名申请，若能够成功注册，则在单独的一行中输出OK，否则在单独的一行中输出作为提示的新用户名。

样例输入

4

abacaba

acaba

abacaba

acab

6

first

first

second

second

third

third

样例输出

OK

OK

abacaba1

OK

OK

first1

OK

second1

OK

third1