2018 阿里巴巴笔试题

一、单项选择题

1、以下函数的时间复杂度是　　（　）

```
1void func(int x,int y, int z){
2if(x<=0)
3printf("%d, %d\n", y, z);
4else
5{
6func(x-1,y+1,z);
7func(x-1,y,z+1);
8}
9}
```

A.O(x*y*z)

B.O(x^2*y^2)

C.O(2^x)

D.O(2^x*2^y*2^z)

E.O(x!)

F.O((x*y*z)!)

参考答案：C

2、在一台 64 位的计算机上，以下哪段 C 语言代码与代码(x[2]+4)[3]等价（x 的类型是 int **）　（　　）

A.*((*(x+16))+28)

B.*((*(x+2))+7)

C.**(x+28)

D.*(((*x)+2)+7)

E.*(((*x)+16)+28)

F.**(x+9)

参考答案：B

3、关于 ios 和 Android 应用以下描述错误的是（）

A.ios 和 Android 应用界面都可以通过 IDE 可视化界面拖拽完成布局，也可以在运行时通过代码布局

B.Objective C 的 ARC 和 Java 的 GC，都是一种运行时内存管理机制

C.ios 和 Android 应用都可以发布或接收通知来进行跨进程通信

D.Ios 和 Android 应用都在调用某些系统功能如相机时，需预先拥有相应权限

E.ios 和 Andriod 应用都拥有各自独立、安全隔离的文件空间

F.ios 和 Andriod 应用都可以注册自定义 URL　Scheme

参考答案：C

4、堆栈中有元素 abcdef，每次出栈可以选择一个或者两个元素栈，当有两个元素出栈时可以选择其中一个重新入栈，则所有元素为空，那么可能的出栈方式有（ ）种？

A.23

B.22

C.21

D.20

E.19

F.18

参考答案：C

5、下列关于 linux 中 kernel space 和 user space 描述错误的是（）

A.user space 不能直接对文件进行写操作

B.程序代码能手动指定在哪个 space 中运行

C.user space 不能直接创建进程

D.user space 和 kernel space 的运行空间是相互隔离的

E.Kernel space 可以执行任意系统命令

F.user space 中运行出错不会影响 kernel space

参考答案：B

6、请阅读下面代码，计算运行结果：

```
public class C{ static class A{
} static class B extends A{
} public static void main(String[]  args){
ArrayList<A> list  = new ArrayList<A>();
list.add(new B()); method1(list);
} private static void method1(List<?[侯萍 1] supper A> list)
{ for(int i=0;i<list.size();i++){
A a = list.get(0);
}
} }
```

以上程序的运行结果可能出现的是：（ ）

A.list.add(new B())编译报错

B.method1 编译报错

C.A a=list.get(0)编译报错

D.程序正常运行

E.list.add(new B())与 method1(list)都编译报错

F.list.add(new B())与 A a=list.get(0);编译报错

参考答案：C

7、请阅读下面代码，计算运行结果;

```
  public class ThreadTest{
1 private static AtomicInteger atomicInteger=new AtomicInteger();
2 public static void main(String[]  args){
3 A a  =new A();
4 try{
5 atomicInteger.wait();
6 } catch (InterruptedException e){
7 e.printStackTrace();
8 }
9 a.start();
10}
11static class A extends Thread{ <a class="js-nc-card" data-card-
12uid="992988" href="/profile/992988" target="_blank">@Override
13public void run(){
14atomicInteger.notify();
15atomicInteger.lazySet(1);
16System.out.println(atomicInteger.get());
17}
18}
  }</a>
```

以上程序的运行结果是：（）

A.编译报错，有未捕捉的异常

B.程序正常运行后，一直 hold

C.程序正常运行，控制台打印出 1

D.程序正常运行，控制台打印出 0

E.程序编译通过，但运行时报错

F.以上都不对

参考答案：A

8、在一台 6G 内存 Linux 操作系统的机器上，coredump 打开且大小不做限制，执行下面的程序分别会发生什么？

（1）

```
1#include<stdio.h>
2#include<stdlib.h>
3int32_t main(){
4unit64_t size=8*1024*1024*1024L;
5char* a=new char(size);
6*(a+1)='a';
7return 0;
8}
```

（2）

```
1#include<stdio.h>
2#include<stdlib.h>
3int32_t main(){
4unit64_t size=10*1024*1024*1024L;
5char* a=new char(size);
6*(a+1)='a';
7return 0;
8}
```

（3）

```
1#include<stdio.h>
2#include<stdlib.h>
3int32_t main(){
4unit64_t size=7*1024*1024*1024L;
5char* a=new char(size);
6*(a+size-1)='a';
7return 0;
8}
```

A.coredump，coredump，coredump

B.正常，coredump，正常

C.正常，正常，coredump

D.coredump，正常，coredump

E.coredump，正常，正常

F.正常，coredump，coredump

参考答案：A

9、下列程序的输出是（）

```
1 #include<iostream>
2 using namespace std;
3 class A{
4 public:
5 A(int n):m_n(n){}
6 int cal(){
7 int result  = 0,i  = 0,j  = 0;
8 for(int k  = m_n;k>0;k--){
9 if(j>0)
10j  = k*10+j;
11else
12j  = k;
13while (j>=10) {
14int t  = j  % 100;
15j  = j  / 100;
16result  = ((i++ % 2==0) ? result+t:result-t);
17}
18}
19if(j>0)
20result  = ((i++ % 2==0) ? result+j:result-j);
21return result;
22}
23private:
24int m_n;
25};
26int main(){
27A a  = A(101);
28cout<<a.cal()<<endl;
29return 0;
30}
```

A.-80

B.-79

C.0

D.90

E.79

F.80

参考答案：A

10、用 0,1,2,3,4,5 组成一个 4 位数，要求每一位都不一样，请问能组成多少个四位数（ ）

A.240

B.280

C.300

D.360

E.400

F.450

参考答案：C

11、小明有 200 个淘公仔，小梅有 20 个电脑包，每次小明给小梅 6 个淘公仔，小梅就给小明 1 个电脑包，经过多少次交互后，小明手中的淘公仔的个数是小梅手中电脑包数量的 11 倍？（ ）

A.4

B.5

C.6

D.7

E.8

F.9

参考答案：A

200-6n=11(20-n)

解析：n=4

12、以下描述正确的是（）

A.线性规划问题是一个 NP-Hard 问题

B.因为单纯形法可以保证在限步数内收敛，所以是复杂度为多项式级别的算法，用于解决线性规划问题

C.内点法只用于解决线性规划问题

D.线性规划区别于非线性规划的地方在于，其达到最优点的时候不需要满足 K-K-T 优化条件

E.一个可解的线性规划问题的主问题和对偶问题分别达到最优化的时候，最优值一定相等

F.以上都不对

参考答案：B

13、设有一个二维数组 A[m][n]，假设 A[0][1]存放位置在 1601(10)，A[3][3]存放位置在 1648(10)，每个元素占一个空间，问 A[2][2](10)存放在什么位置？脚注(10)表示用 10 进制表示。（ ）

A.1616

B.1617

C.1618

D.1631

E.1632

F.1633

参考答案：E

解析：

3n+2=1848-1601，解得 n=15。

每一行 15 个元素，每个元素占据一个空间，因此 A[2][2]=1601+15+2+1=1632

14、天气预报说明天降水概率是 84%，假设降水和时间无关，请问明天中午 12 点之前就降水的概率是多大？（ ）

A.30%

B.40%

C.50%

D.60%

E.70%

F.80%

正确答案：D

15、

```
1 public class ListParamTest {
2 public static void resetList(List<Integer> dataList) {
3 dataList.subList(2,4).set(0,40);
4 dataList = new ArrayList<Integer>(dataList);
5 dataList.add(50);
6 }
7 public static void setOne(List<Integer> dataList) {
8 dataList.set(3,100);
9 }
10public static void main(String[] args){
11List<Integer> dataList = new
12ArrayList<Integer>(Arrays.asList(10,20,30,null));
13resetList(dataList);
14setOne(dataList);
15int sum  = 0;
16for(Integer v:dataList){
17sum  +=v;
18}
19System.out.println(sum);
```

```
}
```

程序执行后，输出的结果是：

A.160

B.抛出 UnsupportedOperationException 异常

C.抛出 NullPointerException 异常

D.220

E.210

F.170

参考答案：F

16、一个等差数列第 x ，y ，z 三项的值分别时是 y ，z ，x ，试求第 x+y 项和第 z+y 项的差值（ ）

A.-3

B.-2

C.-1

D.0

E.1

F.2

正确答案：D

17.机器学习中，下面哪个方法不是为了防止过拟合的？

A.Batchnorm

B.Dropout

C.Weight decay

D.Dropconnect

E.Early stopping

F.Data augmentation

正确答案：A

18.在关联规则挖掘算法中，有已知如下事务类，支持度 support=0.4，则下列选项不正确的是（）

TID	Items
1	Bread，Milk
2	Bread，Diaper，Beer，Eggs
3	Milk，Diaper，Beer，Coke
4	Bread，Milk，Diaper，Beer，
5	Bread，Milk，Diaper，Coke

A.{Bread,Milk}是频繁项集

B. {Bread,Milk,Beer}是{Bread,Milk}的超集

C. {Bread,Milk}是频繁闭项集

D. {Bread,Milk}是最大频繁项集

E. {Bread,Diaper}是频繁项集

参考答案：D

19.评分卡算法（Score Card）是在金融领域广泛应用的一种评分算法，通过多个维度的评分汇总得到对于一个实体的总体评估，一下说法错误的是（）

A.评分卡的底层分类算法最常用的是逻辑回归算法，因此评分卡是一种相对白盒的算法

B.评分卡算法其中一个重要的数据处理步骤是数据分箱，根据特征取值将数据离散化为若干区间，这种操作能对某些数据异常值进行处理

C.评分卡算法中如果变量之间存在多重共线性，说明可能存在两个变量高度相关，需要进行降维或剔除变量

D.评分卡中对用户分类使用的逻辑回归算法是广义线性回归模型的一种

E.评分卡模型效果的验证可以通过 ROC 曲线来看

F.评分卡中的逻辑回归算法可以用于二分类算法，而不能用于多分类问题

参考答案：A

二、编程题

1、天猫国际每天都会卖出很多跨境商品，用户每次下单可能购买多个商品，购买总数小于 10 件，由于海关规定，每一个进入海关的箱子里面的商品总额不能超过 2000 元（否则不能清关）所以当用户下单总金额超过 2000，必须使用多个箱子分开包装运输；现在为了节约运输成本，希望在满足海关的要求下，能够使用尽可能少的箱子。

注：

每个商品都有自己的单价，有特定的长宽高，所有商品都是长方体

商品可以横放、竖放、侧放，但不用考虑斜放，但是长宽高各项总和必须都要小于等于箱子的长宽高

假定目前天猫国际使用同一种规格的箱子

boxLong,boxWidth,boxHigh

（箱子长，箱子宽，箱子高）

某用户下单买了如下商品

n（商品件数）

item1Price,item1Long,item1With,item1High

item2Price,item2Long,item2With,item2High

item3Price,item3Long,item3With,item3High

item4Price,item4Long,item4With,item4High

...

（商品价格，商品长，商品宽，商品高）

(所有输入类型均为 int 型正整数)

请你算出需要使用最小的箱子数量，可以将这些商品顺利得清关送到消费者手中，如果无解，输出-1

代码模板：

```
import java.lang.reflect.Array; import java.util.Scanner; public class Main {
/**请完成下面这个 process 函数，实现题目要求的功能**/
/**当然，你也可以不按照这个模板来作答，完全按照自己的想法来^-^  **/
private static int process()
{
} public static void main(String args[]){
Scanner scanner  = new Scanner(System.in); boxTemplate.price = CUSTOMS_LIMIT_MONEY_PER_BOX; while (scanner.hasNext()){ boxTemplate.length = scanner.nextInt(); boxTemplate.width = scanner.nextInt(); boxTemplate.height = scanner.nextInt(); int itemNum  = scanner.nextInt(); items = new Model[itemNum]; for(int i=0; i<itemNum; i++){ Model item  = new Model();
item.price = scanner.nextInt();
item.length = scanner.nextInt();
item.width = scanner.nextInt();
item.height = scanner.nextInt(); items[i] = item;
} long startTime  = System.currentTimeMillis(); boxMinNum = Integer.MAX_VALUE;
System.out.println (process());
}
}
```

}

2、在快递公司干线运输的车辆使用中，存在着单边车和双边车的两种使用场景，例如北京中心-杭州中心，两个分拨中心到彼此的单量对等，则可以开双边车（即同一辆车可以往返对开），而当两个中心的对发单量不对等时，则会采用单边车，并且双边车的成本是低于单边车的，即将两辆对开的单边车合并为一辆往返的双边车是能够节省运力成本的

单边车优化原则：

将单边车优化的规则进行可抽象为以下三种（A,B,C 均表示分拨中心）：

规则-1: A-B 单边车，B-A 单边车 优化方案：将 A-B 和 B-A 的两辆单边车合并为双边；

规则-2: A-B 单边车，B-C 单边车，C-A 单边车 优化方案：将 A-B、B-C、C-A 的三辆单边车优化为一辆环形往返车；

规则-3: A-B 单边车，C-A 单边车，B、C 同省 优化方案：当 B、C 同省，将 A-B、C-A 两辆单边优化为一辆环形往返

问题如下：

以某快递公司的实际单边车数据为例（线路 ID 编码;出分拨中心;出分拨中心所在省;到达分拨中心;到达分拨中心所在省；车型；），进行优化，优化的规则参照以上，并且优先级依次降低，合并的时候需要考虑车型（分为 17.5m 和 9.6m 两种）：1、相同车型才能进行合并；2、两辆同方向的 9.6m 可以与一辆 17.5m 的对开车型合并优化 说明：优化输出结果按照规则分类，例如 rule1：2016120001+2016120002 表示将单边车线路 ID 编码为 2016120001 和 2016120002 按照规则 1 合并优化

代码模板：

```
1 public class Main {
2 public static void main(String[] args) {
3 Scanner scanner  = new Scanner(System.in);
4 List<UnilateralLine> lineList  = new ArrayList<UnilateralLine>();
5 while (scanner.hasNextLine()) {
6 String[] options  = scanner.nextLine().split(";");
7 if (options.length  < 5) {
```

```
8 break;
9 }
10lineList.add(new UnilateralLine(options[0], options[1], options[2],
11options[3], options[4], options[5]));
12}
13scanner.close();
14// wirte your code here
15List<String> result  = calculateUnilateral(lineList);
16for (String str  : result) {
17System.out.println(str);
18}
19}
20public static List<String> calculateUnilateral(List<UnilateralLine>
21lineList) {
22List<String> result  = new ArrayList<String>();
23return result;
24}
25public static class UnilateralLine {
26private String id;
27private String sCen;//出发分拨
28private String sPro;//出发省
29private String eCen;//到达分拨
30private String ePro;//到达省
31//9.6m/17.5m
32private String tType;//车型
33public UnilateralLine(String id, String sCen, String sPro, String
34eCen, String ePro,String tType) {
35this.id  = id;this.sCen  = sCen;this.sPro  = sPro;this.eCen  =
36eCen;this.ePro  = ePro;this.tType  = tType;}
37public String getId() {return id;}
38public void setId(String id) {this.id  = id;}
39public String getSCen() {return sCen;}
40public void setSCen(String ePro) {this.ePro  = ePro;}
41public String getSPro() {return sPro;}
42public void setSPro(String sPro) {this.sPro  = sPro;}
43public String getECen() {return eCen;}
44public void setECen(String eCen) {this.eCen  = eCen;}
45public String getEPro() {return ePro;}
46public void setEPro(String ePro) {this.ePro  = ePro;}
  public String getTType() {return tType;}
  public void setTType(String tType) {this.tType  = tType;}
  }
  }
```