



寒假作业

WINTER VACATION HOMEWORK

等不是办法，干才有希望
不走心的努力都是敷衍自己

比特就业课假期作业- Java语法+数据结构+EE+软件测试作业

出题老师:

Java基础选择题: 高博 (day01 - day06第4题) qq: 1262913815

JavaEE选择题: 郑智文 (day06第5题 - day16) qq: 16045413

Java基础编程题+MySQL编程题: 郑智文 (day01 - day13) qq: 16045413

Java编程题: 王磊 (day09-day16) qq: 997689302

软件测试作业: 马绵 (day17-day19) qq: 996402440

作业说明:

- 1、本次作业涵盖内容为Java基础, 数据结构, EE初阶, EE进阶, 测试相关知识点
- 2、如果对试卷上的题目, 或者答案有问题, 可以联系对应老师哦~~
- 3、同学们添加老师时备注: 姓名+比特班级哦~

day01

一、选择题

1、派生类调用基类的构造器必须用到的关键字 (C)

A: this B: final C: super D: static

2、以下哪个接口的定义是正确的 (D)

A: interface B{ void print() { }; } 普通方法默认是抽象方法, 不需要有具体实现

B: interface B{ static void print() ; } 静态方法需要有具体实现

C: abstract interface B extends A1, A2{ abstract void print(){ }; } //A1、A2为已定义的接口 接口不用abstract修饰

D: interface B{ void print(); }

3、关键字super的作用是 (D)

A: 用来访问父类被隐藏的非私有成员变量

B: 用来调用父类中被重写的方法

C: 用来调用父类的构造函数

D: 以上都是

4、将下列 (A、B、C、D) 哪个代码替换下列程序中的【代码】不会导致编译错误 (A)

```
interface Com{
    int M=200;
    int f();
}
class ImpCom implements Com{
    //【代码】
}
```

A: `public int f(){ return 100+M;}`

B: `float f(){ return 100;}`

C: `public double f(){ return 2.6;}`

D: `public abstract int f()`

重写方法，要求参数列表，方法名必须一样，返回值一般是一样，但是如果构成父子类的关系也是可以的，访问修饰符子类权限必须大于等于父类权限

5、默认类型等价表示是哪一项 () **C**

```
public interface IService {String NAME="default";}
```

A: `public String NAME="default";`

B: `public static String NAME="default";`

C: `public static final String NAME="default";`

D: `private String NAME="default";`

二、编程题

1、给你一个整数数组 `nums`，数组中的元素互不相同。返回该数组所有可能的子集（幂集）。解集不能包含重复的子集。你可以按任意顺序返回解集。[O链接](#)【LeetCode题号：78. 子集】【中等】

示例 1：

输入：nums = [1,2,3]

输出：[[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]

示例 2：

输入：nums = [0]

输出：[[],[0]]

提示：

1 <= nums.length <= 10

-10 <= nums[i] <= 10

nums 中的所有元素互不相同

【代码实现】：

```
import java.util.ArrayList;
import java.util.List;

public class Solution {
```

```
public List<List<Integer>> subsets(int[] nums) {
```

```
    }  
}
```

2、给定两个整数 n 和 k ，返回范围 $[1, n]$ 中所有可能的 k 个数的组合。你可以按 任何顺序 返回答案。 [OJ链接](#)
【LeetCode题号：77. 组合】【中等】

示例 1：

输入： $n = 4, k = 2$

输出：

```
[  
  [2,4],  
  [3,4],  
  [2,3],  
  [1,2],  
  [1,3],  
  [1,4],  
]
```

示例 2：

输入： $n = 1, k = 1$

输出：[[1]]

提示：

$1 \leq n \leq 20$

$1 \leq k \leq n$

【代码实现】：

```
import java.util.*;
```

```
public class Solution {  
    public List<List<Integer>> combine(int n, int k) {
```

```
}  
}
```

day02

一、选择题

1、1、以下哪些表达式返回为真【多选】 () A D

```
String a="My field1";  
String b="My field1";  
String c=new String("My field1");  
String d=new String("My field1");
```

A: ^{true} a==b B: ^{false} a==c C: ^{false} c=d D: ^{true} ☒ a.equals(b)

2、此代码片段输出正确的值是 ()A

```
public class CharToString {  
    public static void main(String[] args) {  
        char myChar = 'g';  
        String myStr = Character.toString(myChar); 将字符'g' 转为字符串"g"  
        System.out.println("String is: "+myStr);  
    }  
}
```

A: ☒ String is: g

B: String is: 103

3、执行如下代码段后，变量s1引用的字符串值是 () **A**

```
String s1 = "ABCD";  
String s2 = "1234";  
System.out.println(s1 + s2);
```

A: ☒ ABCD B: 1234 C: ABCD1234 D: 1234ABCD

4、有以下代码片段 (

A

```
String str1="hello";
String str2="he"+ new String("llo");
System.out.println(str1.equals(str2));
```

str1 != str2 但是使用equals比较的是字符串的至，所以是true

请问输出的结果是：

A: ☒ true B: ☐ 都不对 C: ☐ null D: ☐ false

5、关于String、StringBuffer、StringBuilder以下说法错误的是 (C)

A: ☒ StringBuilder运行速度最快

B: ☒ StringBuffer是线程安全的

C: ☒ String的值是可变的 字符串的指向可以改变，但是字符串的值是不可变的

D: ☐ StringBuffer运行速度比String快

二、编程题

1、给定一个不含重复数字的数组 nums，返回其 所有可能的全排列。你可以 按任意顺序 返回答案。[OJ链接](#)
【LeetCode题号：46. 全排列】 【中等】

示例 1：

输入：nums = [1,2,3]

输出：[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

示例 2：

输入：nums = [0,1]

输出：[[0,1],[1,0]]

示例 3：

输入：nums = [1]

输出：[[1]]

提示：

1 <= nums.length <= 6

-10 <= nums[i] <= 10

nums 中的所有整数 互不相同

【代码实现】：

```
import java.util.*;

public class Solution {

    public List<List<Integer>> permute(int[] nums) {
```

```
}  
  
}
```

2、给定一个可包含重复数字的序列 `nums`，**按任意顺序** 返回所有不重复的全排列。[OJ链接](#) 【LeetCode题号：47. 全排列 II】【中等】

示例 1：
输入：nums = [1,1,2]
输出：

```
[  
  [1,1,2],  
  [1,2,1],  
  [2,1,1]  
]
```

示例 2：
输入：nums = [1,2,3]
输出：

```
[  
  [1,2,3],  
  [1,3,2],  
  [2,1,3],  
  [2,3,1],  
  [3,1,2],  
  [3,2,1]  
]
```

提示：
1 <= nums.length <= 8
-10 <= nums[i] <= 10

【代码实现】：

```
import java.util.*;  
  
public class Solution {  
  
    public List<List<Integer>> permuteUnique(int[] nums) {
```

```
}  
}
```

day03

一、选择题

1、关于下面程序 ThisTest.java 运行结果 说法正确的是 () **B**

```
public class ThisTest {  
    public static void main(String args[]) {  
        String x="7";  
        int y = 2;  
        int z = 2;  
        System.out.println(x+y+z);  
    }  
}
```

从左往右计算，进行的是字符串的拼接

A: 11 B: **22** C: 22 D: 程序有编译错误

2、已知String a="a", String b="b", String c=a+b, String d=new String("ab") 以下操作结果为true的是【多选】
() **A D**

A: **(a+b).equals(c)** B: b==c C: c==d D: **c.equals(d)**
true **false** **false** **true**

3、下列Java代码中的变量a、b、c分别在内存的存储区存放 () **C**


```

class A {
    private String a = "aa"; // 成员变量: 堆
    public boolean methodB() { // 静态变量 -> 方法区
        String b = "bb"; // 局部变量 -> 栈区
        final String c = "cc"; // 成员变量 -> 堆
    }
}

```

b是局部引用，在栈上
 c是局部常量，栈区
 原本字符串本身就是不可变的，也就是字符串的内容不可变，利用final修饰之后，c的指向也就不能再修改。但是无关final，c还是一个局部的变量，出了方法就不存在了

A: 堆区、堆区、堆区

B: 堆区、栈区、堆区

C: 堆区、栈区、栈区

D: 堆区、堆区、栈区

4、假设有以下代码String s = "hello"; String t = "hello"; char c [] = {'h','e','l','l','o'}; 下列选项中返回false的语句是 ()

A: s.equals(t); // true
 B: t.equals(c); // 都不是字符串类型，直接会返回false
 C: s==t; // true
 D: t.equals(new String("hello"));

5、能单独和finally语句一起使用的块是 ()

A: try B: catch C: throw D: throws

二、编程题

1、数字 n 代表生成括号的对数，请你设计一个函数，用于能够生成所有可能的并且有效的括号组合。

有效括号组合需满足：左括号必须以正确的顺序闭合。 [OJ链接](#) 【LeetCode题号：22. 括号生成】 【中等】

示例 1:

输入: n = 3

输出: ["((()))","(()())","(())()","()(())","()()()"]

示例 2:

输入: n = 1

输出: ["()"]

提示:

1 <= n <= 8

【代码实现】:

```

import java.util.ArrayList;
import java.util.List;

public class Solution {

    public List<String> generateParenthesis(int n) {
    }
}

```

```
}  
}
```

2、给定一个包含红色、白色和蓝色，一共 n 个元素的数组，原地对它们进行排序，使得相同颜色的元素相邻，并按照红色、白色、蓝色顺序排列。此题中，我们使用整数 0、1 和 2 分别表示红色、白色和蓝色。 [OJ链接](#)
【LeetCode题号：75. 颜色分类】【中等】

示例 1:

输入: `nums = [2,0,2,1,1,0]`

输出: `[0,0,1,1,2,2]`

示例 2:

输入: `nums = [2,0,1]`

输出: `[0,1,2]`

示例 3:

输入: `nums = [0]`

输出: `[0]`

示例 4:

输入: `nums = [1]`

输出: `[1]`

提示:

$n == \text{nums.length}$

$1 \leq n \leq 300$

`nums[i]` 为 0、1 或 2

【代码实现】：

```
import java.util.Arrays;  
  
public class Solution {  
  
    public void sortColors(int[] nums) {
```

```
}  
}
```

day04

一、选择题

1、执行如下代码后输出结果为 ()

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("return value of getValue(): " + getValue());  
    }  
    public static int getValue() {  
        int i = 1;  
        try {  
            i = 4;  
        } finally {  
            i++;  
            return i;  
        }  
    }  
}
```

A: return value of getValue(): 1

B: return value of getValue(): 4

☒ C: return value of getValue(): 5

D: 其他几项都不对

2、try块后必须有catch块 () **B**

A: 正确 B: ~~错误~~ 不是必须的

3、以下关于JAVA语言异常处理描述正确的有【多选】 () **CD**

A: ~~throw~~ 关键字可以在方法上声明该方法要抛出的异常。

B: ~~throws~~ 用于抛出异常对象。

两个刚好说反了

✓ C: try是用于检测被包住的语句块是否出现异常，如果有异常，则捕获异常，并执行catch语句。

✓ D: finally语句块是不管有没有出现异常都要执行的内容。

4、下面代码运行结果是 () **C**

```
public class Test{
    public int add(int a,int b){
        try {
            return a+b; 34 + 9 = 43
        } catch (Exception e) {
            System.out.println("catch语句块");
        } finally{
            System.out.println("finally语句块"); finally是一定会执行的
        }
        return 0; 这个return执行不到
    }
    public static void main(String argv[]){
        Test test =new Test();
        System.out.println("和是: "+test.add(9, 34));
    }
}
```

A: catch语句块 和是: 43

B: 编译异常

✓ C: finally语句块

和是: 43

D: 和是: 43

finally语句块

5、以下说法哪个是正确的【多选】 () **ABCD**

编译时异常

A: IOException在编译时会被发现

这个题考察的点就是常见的编译时异常和运行时异常

运行时异常

B: NullPointerException在编译时不被发现

编译时异常

✓ C: SQLException在编译时会被发现

编译时异常

✓ D: FileNotFoundException在编译时会被发现

二、编程题

1、给定整数数组 `nums` 和整数 `k`，请返回数组中第 `k` 个最大的元素。请注意，你需要找的是数组排序后的第 `k` 个最大的元素，而不是第 `k` 个不同的元素。[OJ链接](#) 【LeetCode题号：215. 数组中的第K个最大元素】【中等】

示例 1:

输入: [3,2,1,5,6,4] 和 `k = 2`

输出: 5

示例 2:

输入: [3,2,3,1,2,4,5,5,6] 和 `k = 4`

输出: 4

提示:

$1 \leq k \leq \text{nums.length} \leq 104$

$-104 \leq \text{nums}[i] \leq 104$

【代码实现】：

```
import java.util.PriorityQueue;

public class Solution {
    public int findKthLargest(int[] nums, int k) {

    }
}
```

2、找出所有相加之和为 `n` 的 `k` 个数的组合。组合中只允许含有 1 - 9 的正整数，并且每种组合中不存在重复的数字。[OJ链接](#) 【LeetCode题号：216. 组合总和 III】【中等】

示例 1:

输入: k = 3, n = 7

输出: [[1,2,4]]

示例 2:

输入: k = 3, n = 9

输出: [[1,2,6], [1,3,5], [2,3,4]]

说明:

所有数字都是正整数。

解集不能包含重复的组合。

【代码实现】：

```
import java.util.ArrayDeque;
import java.util.ArrayList;
import java.util.Deque;
import java.util.List;

public class Solution {

    public List<List<Integer>> combinationSum3(int k, int n) {

    }

}
```

day05

一、选择题

- 1、在异常处理中，如释放资源，关闭数据库、关闭文件应由（ ）语句来完成（ ）
A: try子句 B: catch子句 C: finally子句 D: throw子句
- 2、HASH 函数冲突处理方式不包括以下哪一项：（ ）

A. 开放定址法

B. 链地址法

C. 插~~X~~排序法

D. 公共溢出区法

3、HashMap的数据结构是怎样的 () **C**

A. 数组 B. 链表 C. **数组+链表** D. 二叉树

4、以下程序的运行结果是 () **D**

```
TreeSet<Integer> set = new TreeSet<Integer>();
TreeSet<Integer> subSet = new TreeSet<Integer>();
for(int i=606;i<613;i++){
    if(i%2==0){
        set.add(i);
    }
}
subSet = (TreeSet)set.subSet(608,true,611,true);
set.add(609);
System.out.println(set+" "+subSet);
```

606 608 610 612
608 610
返回608到611之间的元素视图，包含两个边界

A. 编译失败

B. 发生运行时异常

C. [606, 608, 609, 610, 612] [608, 609, 610]

D [606, 608, 609, 610, 612] [608, 610]

5、散列函数有共同的性质，则函数值应当以 () 概率取其值域的每一个值。 () **D**

A. 最大 B. 最小 C. 平均 **D. 同等**

同等概率下，发生冲突的概率才会变小

二、编程题

1、给你一个非负整数数组 nums，你最初位于数组的第一个位置。

数组中的每个元素代表你在该位置可以跳跃的最大长度。

你的目标是使用最少的跳跃次数到达数组的最后一个位置。

假设你总是可以到达数组的最后一个位置。 [O链接](#) 【LeetCode题号：45. 跳跃游戏 II】【中等】

```
0 <= nums[i] <= 1000
```

```
public class Solution {  
  
    public int jump(int[] nums) {  
  
  
  
  
  
  
  
  
  
    }  
}
```

输出: 3

输出: 1

输出: 1

nums 中 只有一个整数 出现 两次或多次， 其余整数均只出现 一次

```
class Solution {  
  
    public int findDuplicate(int[] nums) {
```

C. 数据元素过多

不同的元素，根据hash函数哈希出的地址是一样的

✓ 不同键值的元素对应于相同的存储地址

2、对于一棵二叉排序树：(B) 遍历可以得到有序序列

A. 前序 B. ✓ 中序 C. 后序 D. 都可以

3、枚举 (enum) 属于基本数据类型 (primitive type) () B

A. 正确 B. ✓ 错误

4、以下哪些是lambda表达式的重要特征【多选】 () ABC

✓ A. 不需要声明参数类型，编译器可以统一识别参数值。

✓ B. 一个参数无需定义圆括号，但多个参数需要定义圆括号

✓ C. 如果主体包含了一个语句，就不需要使用大括号

D. 如果主体只有一个表达式返回值则编译器会自动返回值，大括号需要指定表达式返回了一个数值

5、以下代码中可以正确设置客户端请求编码为 UTF-8 的是 () A

✓ A. request.setCharacterEncoding("UTF-8")

B. request.setCharset("UTF-8")

C. request.setContentType("UTF-8")

D. request.setEncoding("UTF-8")

二、编程题

1、给定一个 24 小时制 (小时:分钟 "HH:MM") 的时间列表，找出列表中任意两个时间的最小时间差并以分钟数表示。 [OJ链接](#) 【LeetCode题号：539. 最小时间差】 【中等】

示例 1:

输入: timePoints = ["23:59","00:00"]

输出: 1

示例 2:

输入: timePoints = ["00:00","23:59","00:00"]

输出: 0

提示:

$2 \leq \text{timePoints} \leq 2 * 10^4$

timePoints[i] 格式为 "HH:MM"

【代码实现】:

```
import java.util.Arrays;
import java.util.List;

public class Solution {

    public int findMinDifference(List<String> timePoints) {
```

```
}  
}
```

2、假设你正在爬楼梯。需要 n 阶你才能到达楼顶。

每次你可以爬 1 或 2 个台阶。你有多少种不同的方法可以爬到楼顶呢？

注意：给定 n 是一个正整数。[O链接](#) 【LeetCode题号：70. 爬楼梯】 【简单】

示例 1：

输入：2

输出：2

解释：有两种方法可以爬到楼顶。

1. 1 阶 + 1 阶

2. 2 阶

示例 2：

输入：3

输出：3

解释：有三种方法可以爬到楼顶。

1. 1 阶 + 1 阶 + 1 阶

2. 1 阶 + 2 阶

3. 2 阶 + 1 阶

【代码实现】：

```
public class Solution {  
  
    public int climbStairs(int n) {
```

```
}
```

```
}
```

day07

一、选择题

1、以下关于 Servlet 生命周期说法错误的是 ()

C

A: Servlet 容器根据 Servlet 类的位置加载 Servlet 类，成功加载后，由容器创建 Servlet 的实例

B: 对于每一个 Servlet 实例，init() 方法只被调用一次

C: 当 Servlet 容器接收到客户端请求时，调用 Servlet 的 service() 方法以及 destory() 方法处理客户端请求

D: servlet 的实例是由 servlet 容器创建的，所以实例销毁也由容器完成

处理客户端的请求是由service方法中会去调用doPost，doGet方法去处理HTTP请求

2、以下关于转发和重定向的说法错误的是 ()

C

A: 转发通过 request 的 getRequestDispatcher().forward() 方法即可实现，它的作用是在多个页面交互过程中实现请求数据的共享。

B: 重定向可以理解为是浏览器至少提交了两次请求，它是在客户端发挥作用，通过请求新的地址实现页面转向

C: 转发和重定向都可以共享 request 范围内的数据

对于转发而言，如果想要共享request范围内的数据，那么你的源组件和目标组件必须是在同一个web应用下

D: 转发时客户端的 URL 地址不会发生改变，而重定向时客户端浏览器中显示的是新的 URL 地址。

3、用HTML标记语言编写一个简单的网页，网页最基本的结构是 ()

D

A: <html> <head>...</head> <frame>...</frame> </html>

B: <html> <title>...</title> <body>...</body> </html>

C: <html> <title>...</title> <frame>...</frame> </html>

D: <html> <head>...</head> <body>...</body> </html>

4、以下标记符中，用于设置页面标题的是 ()

A

A: <title> B: <caption> C: <head> D: <html>

5、下面哪一项是换行符标记 ()

A: `<body>` B: `` C: `
` D: `<p>`

二、编程题

1、一个机器人位于一个 $m \times n$ 网格的左上角（起始点在下图中标记为“Start”）。

机器人每次只能向下或者向右移动一步。机器人试图达到网格的右下角（在下图中标记为“Finish”）。

问总共有多少条不同的路径？ [OJ链接](#) 【LeetCode题号：62. 不同路径】【中等】



示例 1:

输入: $m = 3, n = 7$

输出: 28

示例 2:

输入: $m = 3, n = 2$

输出: 3

解释:

从左上角开始，总共有 3 条路径可以到达右下角。

1. 向右 -> 向下 -> 向下
2. 向下 -> 向下 -> 向右
3. 向下 -> 向右 -> 向下

示例 3:

输入: $m = 7, n = 3$

输出: 28

示例 4:

输入: $m = 3, n = 3$

输出: 6

提示:

$1 \leq m, n \leq 100$

题目数据保证答案小于等于 $2 * 10^9$

【代码实现】:

```
public class Solution {  
  
    public int uniquePaths(int m, int n) {
```

```
}  
}
```

2、给定一个三角形 `triangle`，找出自顶向下的最小路径和。

每一步只能移动到下一行中相邻的结点上。**相邻的结点** 在这里指的是 **下标与上一层结点下标相同或者等于上一层结点下标 + 1** 的两个结点。也就是说，如果正位于当前行的下标 `i`，那么下一步可以移动到下一行的下标 `i` 或 `i + 1`。[OJ链接](#) 【LeetCode题号：120. 三角形最小路径和】【中等】

示例 1:

输入: `triangle = [[2],[3,4],[6,5,7],[4,1,8,3]]`

输出: 11

解释: 如下面简图所示:

```
  2  
 3 4  
6 5 7  
4 1 8 3
```

自顶向下的最小路径和为 11 (即, $2 + 3 + 5 + 1 = 11$)。

示例 2:

输入: `triangle = [[-10]]`

输出: -10

提示:

$1 \leq \text{triangle.length} \leq 200$

$\text{triangle}[0].\text{length} == 1$

$\text{triangle}[i].\text{length} == \text{triangle}[i - 1].\text{length} + 1$

$-104 \leq \text{triangle}[i][j] \leq 104$

【代码实现】：

```
import java.util.Arrays;  
import java.util.List;  
  
public class Solution {  
  
    public int minimumTotal(List<List<Integer>> triangle) {
```

```
}  
}
```

day08

一、选择题

- 1、关于超链接，以下说法正确的是 ()
A: 不同网页上的图片或文本可以链接到同一网页或网站
B: 不同网页上的图片或文本只能链接到同一网页或网站
C: 同一网页上被选定的一个图片或一处文本可以同时链接到几个不同网站
D: 同一网页上图片或文本不能链接到同一书签
- 2、使用CSS时，需要遵从一定的规范。下面选项中，CSS样式书写正确的是 ()
A: `h1{font: 12px;}` B: `h1[font_size:12px;]`
C: `h1(font: size;)` D: `h1{font-size: 12px;}`
- 3、在CSS中，提供了字体样式属性来控制网页中的字体。下面的字体样式设置正确的是 ()
A: `{font-family: 黑体;}` B: `{font-family: "黑体";}`
C: `{fontFamily: "黑体";}` D: `{font-Family: "黑体";}`
- 4、在CSS中，用于设置首行文本缩进的属性是 ()
A: `text-decoration` B: `text-align` C: `text-transform` D: `text-indent`
- 5、CSS样式 `background-position: -5px 10px` 代表的意义是 ()
A: 背景图片向左偏移5px，向下偏移10px
B: 背景图片向左偏移5px，向上偏移10px
C: 背景图片向右偏移5px，向下偏移10px
D: 背景图片向右偏移5px，向上偏移10px

二、编程题

1、给你一个整数数组 `nums`，请你找出一个具有最大和的连续子数组（子数组最少包含一个元素），返回其最大和。

子数组 是数组中的一个连续部分。[O链接](#) 【LeetCode题号：53. 最大子数组和】 【中等】

示例 1:

输入: nums = [-2,1,-3,4,-1,2,1,-5,4]

输出: 6

解释：连续子数组 [4,-1,2,1] 的和最大，为 6。

示例 2:

输入: nums = [1]

输出: 1

示例 3:

输入: nums = [5,4,-1,7,8]

输出: 23

提示：

```
1 <= nums.length <= 10^5
```

$$-10^4 \leq \text{nums}[i] \leq 10^4$$

【代码实现】：

```
public class Solution {  
  
    public int maxSubArray(int[] nums) {  
  
  
  
  
  
  
  
  
  
}
```

2、你是一个专业的小偷，计划偷窃沿街的房屋。每间房内都藏有一定的现金，影响你偷窃的唯一制约因素就是相邻的房屋装有相互连通的防盗系统，如果两间相邻的房屋在同一晚上被小偷闯入，系统会自动报警。

给定一个代表每个房屋存放金额的非负整数数组，计算你 不触动警报装置的情况下，一夜之内能够偷窃到的最高金额。[OJ链接](#)【LeetCode题号：198. 打家劫舍】【中等】

示例 1:

输入: [1,2,3,1]

输出: 4

解释：偷窃 1 号房屋(金额 = 1)，然后偷窃 3 号房屋(金额 = 3)。

偷窃到的最高金额 = 1 + 3 = 4。

示例 2:

输入: [2,7,9,3,1]

输出: 12

解释：偷窃 1 号房屋（金额 = 2），偷窃 3 号房屋（金额 = 9），接着偷窃 5 号房屋（金额 = 1）。

偷窃到的最高金额 = $2 + 9 + 1 = 12$ 。

提示：

```
1 <= nums.length <= 100
```

```
0 <= nums[i] <= 400
```

【代码实现】：

```
public class Solution {  
  
    public int rob(int[] nums) {
```

day09

一、选择题

1、当使用 TCP 协议编程时，下列问题哪个是必须由程序员考虑和处理的（）

A: 乱序数据包的重传

B: 数据传输过程中的纠错

C: 网络拥塞处理

☒ D: 发送数据的格式和应用层协议

2、TCP 断开连接的四次挥手中，第四次挥手发送的包会包含的标记，最正确的描述是 () **C**

A: FIN B: FIN,PSH ☒ C: ACK D: FIN,ACK

客户端
FIN

服务器

3、某浏览器发出的HTTP 请求报文如下:

ACK

ACK
FIN

```
GET /index.html HTTP/1.1
HOST www.test.edu.cn
Connection: Close
Cookie: 123456
```

下列叙述中，错误的是 () **C**

☒ A: 该浏览器请求浏览 index.html

☒ B: index.html 存放在 www.test.edu.cn 上

☒ C: 该浏览器请求使用持续连接 没有keep alive字段

☒ D: 该浏览器曾经浏览过 www.test.edu.cn

4、分析下面的JavaScript代码

```
var x=11; var y="number"; var m=x+y;
```

经过运算后 m 的值为 () **A**

☒ A: 11number B: number C: 11 D: 程序报错

5、文件下载时，用于设定实体内容的MIME类型的响应头是 () **B**

A: Content-Disposition ☒ B: Content-Type

C: ContentEncoding D: ContentAttachment

二、编程题

1、某网站包含两个表，Customers 表和 Orders 表。编写一个 SQL 查询，找出所有从不订购任何东西的客户。 [链接](#) 【LeetCode题号：183. 从不订购的客户】 【简单】

Customers 表:

```
+----+-----+
| Id | Name |
+----+-----+
| 1  | Joe  |
| 2  | Henry|
| 3  | Sam  |
| 4  | Max  |
+----+-----+
```

Orders 表:

Id	CustomerId
1	3
2	1

例如给定上述表格，你的查询应返回:

Customers
Henry
Max

【代码实现】:

比特就业课

2、求 $1+2+\dots+n$ ，要求不能使用乘除法、for、while、if、else、switch、case等关键字及条件判断语句（A? B:C）。[O链接](#)【剑指 Offer 64. 求 $1+2+\dots+n$ 】【难度：中等】

示例 1:

输入: $n = 3$

输出: 6

示例 2:

输入: $n = 9$

输出: 45

限制: $1 \leq n \leq 10000$

【代码实现】

```
class Solution {
    public int sumNums(int n) {

    }
}
```

day10

一、选择题

1、以下选项中，哪个不属于Java运行时数据区域（）

A: ☒ 工作内存 B: 虚拟机栈 C: 堆 D: 方法区

Java运行时数据区域：堆，虚拟机栈，方法区，程序计数器，本地方法栈
这里的工作内存指的是计算机的

2、关于JVM，以下说法不正确的是（）（多选）

工作内存是一个大的概念，在多线程中，我们在JMM模型中，工作内存指的是CPU的寄存器和高速缓存

A: 启动一个Java进程，本质是在Java虚拟机中运行 class 字节码

B: Java程序只能运行在 HotSpot 虚拟机中

C: Java虚拟机可以自动管理Java进程的内存

D: JDK 中包含了 JVM，但是 JRE 中不包含

JDK = JRE + 开发工具集
JRE = JVM + 核心类库

3、Java程序，在什么时候会执行类加载（）（多选）

A: ☒ 创建类的实例

B: 使用 java.exe 命令来运行某个主类的 main 方法

C: ☒ 使用反射来执行某个类的静态方法

D: ☒ 使用 new 创建子类的实例对象时，父类需要先执行类加载

4、在Java程序中，垃圾回收如何判断一个对象是否可以回收（）

A: ☒ 基于引用计数算法 B: ☒ 基于可达性分析算法 C: 基于复制算法 D: 基于标记-整理算法

JVM采用的是可达性的分析

5、垃圾回收的算法有（）（多选）

A: ☒ 标记-清除算法 B: ☒ 复制算法 C: ☒ 标记-整理算法 D: ☒ 分代收集算法

二、编程题

1、编写一个 SQL 查询，查找 Person 表中所有重复的电子邮箱。[OJ链接](#) 【LeetCode题号：182. 查找重复的电子邮箱】 【简单】

示例：

+----+-----+

```
| Id | Email |  
+---+-----+  
| 1 | a@b.com |  
| 2 | c@d.com |  
| 3 | a@b.com |  
+---+-----+
```

根据以上输入，你的查询应返回以下结果：

```
+-----+  
| Email |  
+-----+  
| a@b.com |  
+-----+
```

说明：所有电子邮箱都是小写字母。

【代码实现】：

2、给你一个链表，两两交换其中相邻的节点，并返回交换后链表的头节点。你必须在不修改节点内部的值的情况下完成本题（即，只能进行节点交换）。[O链接](#) **【LeetCode题号24. 两两交换链表中的节点】** **【难度：中等】**

示例 1:
输入：head = [1,2,3,4]
输出：[2,1,4,3]

示例 2:
输入：head = []
输出：[]

示例 3:
输入：head = [1]
输出：[1]

【代码实现】：

```
class Solution {  
    public ListNode swapPairs(ListNode head) {
```

```
}  
}
```

day11

一、选择题

1、以下关于新生代的说法，不正确的是 () **B**

A: Java对象优先分配在新生代中

B: 新生代是方法区中的一块

回收的是对象，所以针对的是堆

C: 新生代包含一块 Eden 区和两个 Survivor 区 新生代分为伊甸区和生存区

D: 新生代垃圾回收速度比老年代快 新生代的扫描次数会比老年代多很多，不断地GC，最终活下来的新生代中的对象会进入老年代

2、关于 CMS 垃圾收集器，以下说法正确的是 () **D**

老年代收集器，并发GC

A: CMS 是所有垃圾收集器中最好的

B: 它是一款单线程的垃圾收集器

C: 对于性能要求高的Java应用，可以使用 CMS 垃圾收集器

☒ D: CMS 是老年代的垃圾收集器

3、关于 volatile 的说法，正确的是 ()

☒ A: volatile 可以保证变量在多线程中任何操作都是安全的

☒ B: volatile 也可以修饰方法，表示该方法是线程安全的

☒ C: volatile 可以保证变量在多线程中“读”操作是安全的 解决的是内存可见性问题，保证每次都会去内存中读取数据

☒ D: volatile 也可以满足原子性

4、关于 synchronized 的说法，正确的是 () (多选) **ACD**

☒ A: synchronized 可以使用在静态方法和实例方法上，也可以使用在代码块上

B: synchronized 一定可以保证多线程执行某段代码是安全的

☒ C: synchronized 是基于对象头来进行加锁的

☒ D: synchronized 的代码，在一个时间上，只能有一个线程执行

5、关于程序和进程，以下说法正确的是 () **B**

A: 两者是相同的

B:  程序是静态的可执行文件，进程是运行中的程序 运行起来的程序才能叫做进程

C: `Main.java` 是一个程序

D: `Main.class` 是一个程序

二、编程题

1、某城市开了一家新的电影院，吸引了很多人过来看电影。该电影院特别注意用户体验，专门有个 LED 显示板做电影推荐，上面公布着影评和相关电影描述。

作为该电影院的信息部主管，您需要编写一个 SQL 查询，找出所有影片描述为非 `boring` (不无聊) 的并且 `id` 为奇数的影片，结果请按等级 `rating` 排列。[O链接](#) 【LeetCode题号：620. 有趣的电影】【简单】

例如，下表 cinema:

id	movie	description	rating
1	War	great 3D	8.9
2	Science fiction		8.5
3	Irish	boring	6.2
4	Ice song	Fantasy	8.6
5	House card	Interesting	9.1

对于上面的例子，则正确的输出是为：

id	movie	description	rating
5	House card	Interesting	9.1
1	War	great 3D	8.9

【代码实现】：

2、给你一个整数数组 `nums`，除某个元素仅出现一次外，其余每个元素都恰出现三次。请你找出并返回那个只出现了一次的元素。[O链接](#)【LeetCode题号137. 只出现一次的数字II】【难度：中等】

示例 1:

输入: `nums = [2,2,3,2]`

输出: 3

示例 2:

输入: `nums = [0,1,0,1,0,1,99]`

输出: 99

【代码实现】:

```
class Solution {  
    public int singleNumber(int[] nums) {
```

```
    }  
}
```

day12

一、选择题

1、进程的状态有 () (多选) **BC**

A: 创建态 ☒ B: 运行态 ☒ C: 阻塞态 ☒ D: 终止态 ☐ 进程的状态: 就绪, 运行, 阻塞

2、Ajax应用程序的优势在于 () (多选) **ABC**

☒ A: 通过异步模式, 提升了用户体验

☒ B: 优化了浏览器和服务器之间的传输, 减少不必要的数据往返, 减少了带宽占用

☒ C: 通过 Ajax 技术可实现只更新局部页面而不需要刷新整个页面

D: 不使用 Ajax 技术, 就不能实现文件上传和下载

3、tomcat 默认的端口是 () **B**

A: 80 ☒ B: 8080 ☒ C: 443 ☐ D: 8888 ☐

4、tomcat 中，webapps 目录的作用是()

A: 部署web应用 B: 存放日志文件 C: 存放配置文件 D: 存放可执行程序

5、DNS 协议的作用是()

A: 根据 IP 地址获取 MAC 地址

B: 根据域名获取 IP 地址

C: 将私网 IP 转换为公网 IP

D: 将私网端口地址转换为公网端口

二、编程题

1、Employee 表包含所有员工信息，每个员工有其对应的 Id, salary 和 department Id。

```
+-----+-----+-----+
| Id | Name | Salary | DepartmentId |
+-----+-----+-----+
| 1 | Joe | 70000 | 1 |
| 2 | Jim | 90000 | 1 |
| 3 | Henry | 80000 | 2 |
| 4 | Sam | 60000 | 2 |
| 5 | Max | 90000 | 1 |
+-----+-----+-----+
```

Department 表包含公司所有部门的信息。

```
+-----+
| Id | Name |
+-----+
| 1 | IT |
| 2 | Sales |
+-----+
```

编写一个 SQL 查询，找出每个部门工资最高的员工。对于上述表，您的 SQL 查询应返回以下行（行的顺序无关紧要）。[OJ链接](#) 【LeetCode题号：184. 部门工资最高的员工】 【中等】

```
+-----+-----+-----+
| Department | Employee | Salary |
+-----+-----+-----+
| IT | Max | 90000 |
| IT | Jim | 90000 |
| Sales | Henry | 80000 |
+-----+-----+-----+
```

解释：

Max 和 Jim 在 IT 部门的工资都是最高的，Henry 在销售部的工资最高。

【代码实现】：

2、给定一个字符串，请将字符串里的字符按照出现的频率降序排列。[OJ链接](#)【LeetCode题号451. 根据字符出现频率排序】【难度：中等】

示例 1:

输入:"tree"

输出:"eert"

解释:'e'出现两次，'r'和't'都只出现一次。因此'e'必须出现在'r'和't'之前。此外，"eetr"也是一个有效的答案。

示例 2:

输入:"cccaaa"

输出:"cccaaa"

解释:'c'和'a'都出现三次。此外，"aaaccc"也是有效的答案。注意"cacaca"是不正确的，因为相同的字母必须放在一起。

示例 3:

输入:"Aabb"

输出:"bbAa"

解释:此外，"bbaA"也是一个有效的答案，但"Aabb"是不正确的。注意'A'和'a'被认为是两种不同的字符。

【代码实现】：

```
class Solution {  
    public String frequencySort(String s) {
```

```
}  
}
```

day13

一、选择题

1、以下哪些属于 TCP 协议的可靠性机制 () (多选) **ABC**

A: 确认应答机制 **✓** B: 超时重传机制 **✓** C: 流量控制 **✓** D: 滑动窗口
基于滑动窗口的可靠机制
效率机制

2、TCP 和 UDP 的区别有 () (多选) **BC**

A: TCP 是传输层协议, UDP 是网络层协议 **✗**

B: **✓** TCP 是有连接的可靠协议, UDP 是无连接的不可靠协议

C: **✓** TCP 是面向字节流, UDP 是面向数据报

D: TCP 既有发送缓冲区, 也有接收缓冲区; UDP 只有接收缓冲区

3、网络通信中, 出现粘包问题, 可以采用以下哪种方案来解决 () (多选) **BD**

A: **✗** 使用 TCP 协议

B: **✓** 使用 UDP 协议 面向数据报, 所以不存在粘包问题

C: **✗** 换一个速度快的网络

D: **✓** 明确应用层数据包的边界

4、线程池的作用是 () **D**

A: **✓** 降低资源消耗: 通过重用已经创建的线程来降低线程创建和销毁的消耗

B: **✓** 提高响应速度: 任务到达时不需要等待线程创建就可以立即执行

C: **✓** 提高线程的可管理性: 线程池可以统一管理、分配、调优和监控

D: 以上都是

5、以下关于 CAS 的说法中, 正确的是 () **B**

A: CAS 是悲观锁 **✗** B: **✓** CAS 是乐观锁 C: CAS 可以替代 synchronized D: CAS 可以替代 volatile

CAS使用面比较小, 但是效率高

二、编程题

1、编写一个 SQL 查询, 查找所有至少连续出现三次的数字。返回的结果表中的数据可以按 任意顺序 排列。 [O链接](#)
【LeetCode题号: 180. 连续出现的数字】【中等】

表: Logs

+-----+-----+	
Column Name	Type
+-----+-----+	
id	int
num	varchar
+-----+-----+	

id 是这个表的主键。

查询结果格式如下面的例子所示:

Logs 表:

+---+---+	
Id	Num
+---+---+	
1	1
2	1
3	1
4	2
5	1
6	2
7	2
+---+---+	

Result 表:

+-----+-----+	
ConsecutiveNums	
+-----+-----+	
1	
+-----+-----+	

1 是唯一连续出现至少三次的数字。

【代码实现】:

2、给定一个字符串 `S`，通过将字符串 `S` 中的每个字母转变大小写，我们可以获得一个新的字符串。返回所有可能得到的字符串集合。[OJ链接](#)【LeetCode题号784. 字母大小写全排列】【难度：中等】

示例：

输入：S = "a1b2"

输出：["a1b2", "a1B2", "A1b2", "A1B2"]

输入：S = "3z4"

输出：["3z4", "3Z4"]

输入：S = "12345"

输出：["12345"]

【代码实现】：

```
class Solution {  
    public List<String> letterCasePermutation(String S) {
```

```
    }  
}
```

day14

一、选择题

1、下面说法正确的是 ()

A: JAVA中线程是非抢占式的

抢占式执行

B: JAVA中的线程不可以共享数据 ~~堆~~，方法区是共享的

☒ C: 每个JAVA程序都至少有一个线程，即主线程 一个进程至少包含一个线程

D: JAVA中的线程不可以共享代码

2、建立和关闭一个 tcp，一共会产生的握手次数是 () **A**

☒ A: 7 B: 6 C: 3 D: 5

这里的握手指的应该是交互次数

3、在计算机网络中，TCP和UDP协议的相似之处是 () **D**

A: 面向非连接的协议

并且都是全双工

B: 面向连接的协议

C: 其余选项都不对

☒ D: 传输层协议

4、已知一个进程绑定的端口号，查询该进程的命令是 () **C**

A: ps B: find ☒ C: netstat D: show

一个端口只能绑定一个进程，但是一个进程可以绑定多个端口

netstat -aon|findstr "端口号"

5、以下协议属于应用层的有 () **D**

A: TCP B: IP C: ARP ☒ D: FTP

二、编程题

1、给定一个由 '(' 和 ')' 括号组成的字符串 S，我们需要添加最少的括号 ('(' 或是 ')'，可以在任何位置)，以使得到的括号字符串有效。从形式上讲，只有满足下面几点之一，括号字符串才是有效的：

- 它是一个空字符串，或者
- 它可以被写成 AB (A 与 B 连接)，其中 A 和 B 都是有效字符串，或者
- 它可以被写作 (A)，其中 A 是有效字符串。

给定一个括号字符串，返回为使结果字符串有效而必须添加的最少括号数。 [OJ链接](#) 【LeetCode题号921. 使括号有效的最少添加】 【难度：中等】

示例 1:

输入: "())"

输出: 1

示例 2:

输入: "((("

输出: 3

示例 3:

输入: "()"

输出: 0

示例 4:

输入: "())(("

输出: 4

【代码实现】：

。如果正整数 i 满足 $n \% i == 0$ ，那么我们就说正整数 i 是 n 的因子。我们把所有可以整除 n 的正整数按从小到大排序。请你返回第 k 个因子。如果 n 的因子数少于 k ，返回 -1 。【难度：中等】

。如果正整数 i 满足 $n \% i == 0$ ，那么我们就说正整数 i 是 n 的因子。我们把所有可以整除 n 的正整数按从小到大排序。请你返回第 k 个因子。如果 n 的因子数少于 k ，返回 -1 。【难度：中等】

解释：因子列表包括[1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 100, 125, 200, 250, 500, 1000]。

【代码实现】：

```
class Solution {
    public int kthFactor(int n, int k) {
```

```
}  
}
```

EE进阶作业

day15

一、选择题

1、已知项目中定义了如下Controller:

```
@Controller  
@RequestMapping("/user")  
public class UserController {  
    @RequestMapping(path = "/otp/{phone}", method = RequestMethod.GET)  
    @ResponseBody  
    public String getOTP(@PathVariable("phone") String phone) {  
        ...  
    }  
}
```

以下URL中，可以正确访问UserController的getOTP方法的是 ()

A: /user/otp B: /otp/user C: /user/otp/13912345678 D: /otp/13912345678/user

2、若要在Controller中声明一个访问路径为"/set"，并且只能响应POST请求的方法，则下列注解中正确的是 ()

A: @RequestMapping("/set")

B: @RequestMapping(path = "/set")

C: @RequestMapping(path = "/set", method = RequestMethod.POST)

D: 其他选项都正确

3、下列选项中，哪一个不是Spring MVC的核心组件 ()

A: DispatcherServlet B: SpringFactoriesLoader C: HandlerMapping D: ModelAndView

4、下列选项中，哪一项不是Spring AOP支持的通知类型 ()

A: 前置通知 B: 后置通知 C: 织入通知 D: 异常通知

5、Spring Bean 的默认作用范围是 ()

A: singleton B: prototype C: request D: session

二、编程题

1、夏日炎炎，小男孩Tony想买一些雪糕消消暑。商店中新到n支雪糕，用长度为n的数组costs表示雪糕的定价，其中costs[i]表示第i支雪糕的现金价格。Tony一共有coins现金可以用于消费，他想要买尽可能多的雪糕。给你价格数组costs和现金量coins，请你计算并返回Tony用coins现金能够买到的雪糕的最大数量。注意：Tony 可以按任意顺序购买雪糕。[OJ链接](#)【LeetCode题号1833. 雪糕的最大数量】【难度：中等】

示例 1:

输入: costs = [1,3,2,4,1], coins = 7

输出: 4

解释: Tony 可以买下标为 0、1、2、4 的雪糕，总价为 1 + 3 + 2 + 1 = 7

示例 2:

输入: costs = [10,6,8,7,7,8], coins = 5

输出: 0

解释: Tony 没有足够的钱买任何一支雪糕。

示例 3:

输入: costs = [1,6,3,1,2,5], coins = 20

输出: 6

解释: Tony 可以买下所有的雪糕，总价为 1 + 6 + 3 + 1 + 2 + 5 = 18。

【代码实现】：

```
class Solution {
    public int maxIceCream(int[] costs, int coins) {

    }
}
```


3、下列不属于Spring Boot注解的是()

A: @Controller B: @AutoConfigurations C: @SpringBootApplication D: 以上都不属于

4、下列关于Spring AOP的实现方式的说法中，正确的是【多选】()

A: JDK动态代理，是Java提供的动态代理技术，可以在运行时创建接口的代理实例。

B: JDK动态代理，是Java提供的动态代理技术，可以在运行时创建子类的代理实例。

C: CGLib动态代理，采用底层的字节码技术，在运行时创建接口代理的实例。

D: CGLib动态代理，采用底层的字节码技术，在运行时创建子类代理的实例。

5、关于Spring注解，下列说法正确的是【多选】()

A: @Autowired 是Spring提供的注解。

B: @Autowired 是JDK提供的注解。

C: @Resource 是Spring提供的注解。

D: @Resource 是JDK提供的注解。

二、编程题

1、假设有打乱顺序的一群人站成一个队列，数组people表示队列中一些人的属性（不一定按顺序）。每个people[i] = [hi, ki]表示第i个人的身高为hi，前面正好有ki个身高大于或等于hi的人。请你重新构造并返回输入数组people所表示的队列。返回的队列应该格式化为数组queue，其中queue[j] = [hj, kj]是队列中第j个人的属性（queue[0] 是排在队列前面的人）。[O链接](#)【LeetCode题号406. 根据身高重建队列】【难度：中等】

示例 1:

输入: people=[[7,0],[4,4],[7,1],[5,0],[6,1],[5,2]]

输出: [[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]]

解释: 编号为0的人身高为5，没有身高更高或者相同的人排在他前面。编号为1的人身高为7，没有身高更高或者相同的人排在他前面。编号为2的人身高为5，有2个身高更高或者相同的人排在他前面，即编号为0和1的人。编号为3的人身高为6，有1个身高更高或者相同的人排在他前面，即编号为1的人。编号为4的人身高为4，有4个身高更高或者相同的人排在他前面，即编号为0、1、2、3的人。编号为5的人身高为7，有1个身高更高或者相同的人排在他前面，即编号为1的人。因此[[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]]是重新构造后的队列。

示例 2:

输入: people=[[6,0],[5,0],[4,0],[3,2],[2,2],[1,4]]

输出: [[4,0],[5,0],[2,2],[3,2],[1,4],[6,0]]

【代码实现】:

```
class Solution {  
    public int[][] reconstructQueue(int[][] people) {
```

```
}  
}
```

2、请你设计一个支持下述操作的栈。实现自定义栈类CustomStack:

- CustomStack(int maxSize): 用maxSize初始化对象, maxSize是栈中最多能容纳的元素数量, 栈在增长到maxSize之后则不支持push操作。
- void push(int x): 如果栈还未增长到maxSize, 就将x添加到栈顶。
- int pop(): 弹出栈顶元素, 并返回栈顶的值, 或栈为空时返回-1。
- void inc(int k, int val): 栈底的k个元素的值都增加val, 如果栈中元素总数小于k, 则栈中的所有元素都增加val。

[O链接](#) 【LeetCode题号1381. 设计一个支持增量操作的栈】 【难度: 中等】

输入:

```
["CustomStack","push","push","pop","push","push","push","increment","increment","pop","pop","pop","pop"]  
[[3],[1],[2],[2],[3],[4],[5,100],[2,100],[1],[1],[1],[1]]
```

输出:

```
[null,null,null,2,null,null,null,null,null,103,202,201,-1]
```

解释:

```
CustomStack customStack = new CustomStack(3); // 栈是空的 []  
customStack.push(1);           // 栈变为 [1]  
customStack.push(2);           // 栈变为 [1, 2]  
customStack.pop();              // 返回 2 --> 返回栈顶值 2, 栈变为 [1]  
customStack.push(2);           // 栈变为 [1, 2]  
customStack.push(3);           // 栈变为 [1, 2, 3]  
customStack.push(4);           // 栈仍然是 [1, 2, 3], 不能添加其他元素使栈大小变为 4  
customStack.increment(5, 100);  // 栈变为 [101, 102, 103]  
customStack.increment(2, 100);  // 栈变为 [201, 202, 103]  
customStack.pop();              // 返回 103 --> 返回栈顶值 103, 栈变为 [201, 202]  
customStack.pop();              // 返回 202 --> 返回栈顶值 202, 栈变为 [201]  
customStack.pop();              // 返回 201 --> 返回栈顶值 201, 栈变为 []  
customStack.pop();              // 返回 -1 --> 栈为空, 返回 -1
```

【代码实现】:

```
class CustomStack {  
  
    public CustomStack(int maxSize) {
```

return k, int val) {

day17

1、下面叙述中错误的是（ ）

- 2、下面说法正确的选项是（ ）

- A. β 测试先于 α 测试执行
- B. β 测试是在受控环境下进行的测试

- C. 模块测试完成之后开始α测试
- D. β测试需要开发者在现场配合测试
- 3、研发需要对自己的代码审核，检查代码的正确性，这个过程我们称为（ ）
- A. 代码走查 B. 单元测试 C. Main函数校验代码 D. 输入输出参数检验
- 4、哪种测试模型把测试过程作为需求分析、概要设计、详细设计及编码之后的阶段（ ）
- A. V模型 B. W模型 C. H模型 D. X模型
- 5、单元测试的测试用例主要根据（ ）的结果来设计（ ）
- A. 需求分析 B. 源程序 C. 概要设计 D. 详细设计

二、问答题

1、什么是软件测试？

2、软件测试的流程？

3、如何根据需求来设计测试用例？

4、什么是测试用例，测试用例的要素是哪一些？

5、什么是软件错误（BUG）？描述BUG的要素有哪些？

6、简述敏捷开发模型Scrum工作流程？

7、软件测试模型有哪些？分别有什么特点？

8、黑盒测试设计测试用例的方法有哪些？

day18

一、选择题

- 1、在一个软件测试用例中，有一个字符串数据，若首字符必须是C和D，其余字符必须是0-9的数字，在此情况下是正确的，若首字符不正确，则输出message1，若其余字符不正确，则输出message2。这种情况下最好采用（ ）
- A. 边界值分析 B. 等价类划分 C. 因果图 D. 错误推测法
- 2、某程序要求每次输入只能是正整数，并且每次输入的数值要求必须是100的倍数且小于等于500，则下列哪个是正确的无效等价类（ ）
- A. (0, 100)、(100, 200)、(200, 300)、(300, 400)、(400, 500)、(500, +∞)
- B. (500, +∞)
- C. (500, +∞)、任意大于0小于500的非100倍数的整数
- D. (-∞, 100)、(100, 200)、(200, 300)、(300, 400)、(400, 500)、(500, +∞)
- 3、如果某测试用例集实现了某软件的路径覆盖，那么它一定同时实现了该软件的（ ）
- A. 条件覆盖 B. 判定/条件覆盖 C. 组合覆盖 D. 判定覆盖
- 4、Beta 测试是验收测试的一种。请判断这句话的正确与否（ ）
- A. T B. F
- 5、单元测试的测试效率要高于UI界面测试。请判断这句话的正确与否（ ）
- A. T B. F

二、问答题

- 1、等价类法设计测试用例的思想是什么？

- 2、因果图法设计测试用例的步骤？

- 3、按照开发阶段划分测试流程，都有哪些阶段？每一个阶段的测试内容是什么？

- 4、按照是否查看代码划分，软件测试分为哪几种测试类型？

- 5、什么是黑盒测试？

6、什么是冒烟测试？

7、禅道是一款什么工具？

8、webdriver API定位元素的方式都有哪些？

day19

一、选择题

1、测试工程师在软件测试计划阶段依据（ ）制定指定测试进度（ ）

- A. 需求说明书
- B. 概要设计说明书
- C. 详细设计说明书
- D. 单元测试用例

2、以下叙述中，不正确的有【多选】（ ）

- A. 单元测试对源程序中每一个程序单元进行测试，检查各个模块是否正确实现规定的功能，从而发现模块在编码中或算法中的错误。该阶段涉及编码和详细设计文档。
- B. 集成测试是基于软件需求说明书的黑盒测试，是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确
- C. 确认测试主要是检查已实现的软件是否满足需求规格说明书中确定了的各种需求。
- D. 系统测试的主要目的是检查软件单位之间的接口是否正确，主要是针对程序内部结构进行测试，特别是对程序之间的接口进行测试。

3、alpha测试与beta的区别，描述错误的是（ ）

- A. alpha测试是在用户组织模拟软件系统的运行环境下的一种验收测试，由用户或第三方测试公司进行的测试，模拟各类用户行为对即将面市的软件产品进行测试，试图发现并修改错误。
- B. Beta测试是用户公司组织各方面的典型终端用户在日常工作中实际使用beta版本，并要求用户报告异常情况，提出批评意见。
- C. beta测试的环境是不受开发方控制的，谁也不知道用户如何折磨软件，用户数量相对比较多，时间不集中。
- D. beta测试先于alpha测试执行
- 4、下列测试用例设计的方法中，哪些属于黑盒测试方法【多选】（）
- A. 语句覆盖 B. 逻辑覆盖 C. 边界值分析 D. 等价类划分法
- 5、下面描述测试工具的功能正确的有【多选】（）
- A. JMeter: 基于JAVA的压力测试工具，Badboy用来进行脚本的录制
- B. Junit: 白盒测试工具：针对代码测试
- C. LoadRunner:负载压力测试
- D. TestLink: 用例管理工具
- 6、下面哪些属于网游的测试内容【多选】（）
- A. 客户端性能
- B. 服务器端性能
- C. 从运行完 game.exe 打开游戏界面后可进行的各种操作、玩法
- D. 界面

二、问答题

1、编程题：请用Python+selenium编写网易邮箱登陆脚本的测试用例？登陆的条件是输入账户（account）和密码（password），写完脚本之后请大家实战一下，看是否可以成功自动登录，如果不能，想想为什么？

2、什么是数据驱动？使用数据驱动时常用的标签有哪些？

3、unittest中组织测试套件（Test Suite）的方式有哪些？每一种的作用是什么？

4、什么是断言，自动化测试脚本中断言的作用是什么？常用的断言有哪些？

5、常见的性能测试指标都有哪些？分别代表什么意义？

6、什么是基准性能测试？

7、什么是并发测试？

8、什么是压力测试？

9、Loadrunner三个组件是什么，分别有什么作用？

10、Jenkins是一款什么工具？是开源的吗？

比特就业课