

猿辅导公司某研发小组一共有 12 名同学，其中 9 人能做后端开发，6 人能做前端开发。现在要抽调 4 名同学成立项目小组，负责公司的一项“机密”项目。其中 2 名同学做后端开发，2 名同学做前端开发。有多少种选派方法？ C

A. 180      B. 225      C. 423      D. 540

2. 关于数据库，以下描述错误的是(D)

- A. 任意两行的主键值都不相同
- B. 主键列不允许 NULL 值
- C. 表可包含多个唯一约束，但每个表只允许一个主键
- D. 唯一约束列不允许 NULL 值

3. 有一个虚拟存储系统，若进程在内存中占 3 页，开始内存为空。若采用 LRU 页面淘汰算法，当执行如下访问页号序列 [1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 5, 4] 会产生多少次缺页？

B

A. 9 次      B. 10 次      C. 11 次      D. 12 次

4. 假设解决某问题的算法复杂度为  $F(n)$ ，其中  $n$  为问题规模，该问题总是可以通过  $g(n)$  的代价划分为 2 个规模为  $n/2$  的同类子问题，即  $F(n)=2F(n/2)+g(n)$ ，当  $g(n)$  为  $O(n)$ ，时  $F(n)$  为多少？

C

A.  $O(\log n)$     B.  $O(n)$     C.  $O(n \log n)$     D.  $O(n^2)$

5. 关于 HTTP 协议，下列说法错误的是(C)

- A. HTTP 是无状态协议
- B. HTTP 是应用层协议
- C. HTTP 是基于二进制的协议
- D. HTTP 状态码 200 表示成功

6. 关于进程和线程，下列说法错误的是(B)

- A. 同一进程内的线程共享内存等系统资源
- B. 进程之间不能通信
- C. 进程有独立的虚拟地址空间
- D. 线程可以创建另外一个线程

7. 关于 TCP 协议，下列说法错误的是(B)

- A. TCP 是面向连接的协议
- B. TCP 是基于数据报文的协议
- C. TCP 提供了拥塞控制的机制
- D. TCP 属于传输层协议

8. 将 A B C D 四个元素次序进栈（中间可能有出栈操作，例如 A 进栈后出栈，B 再进栈），则可能的出栈序列是(B)

A. C A D B      B. B D C A  
C. C D A B      D. D A B C

9. 某天猿辅导 HR 组织大家去漂流，早上，参加团建的同学都到齐了，并且按到达公司的先后顺序排好队了。由于员工太多，一个大巴车坐不下，需要分多个车，车是足够的，但所有人需要按一定顺序上车，按如下规则安排上车的顺序：

假设大巴车容量为  $m$ ，从队首开始，每  $m$  个人分成一个小组，每个小组坐一辆车。同时只有一个车打开车门供员工上车。小组之间按从队尾到队首顺序依次上车，同一小组内先到的同学先上，求所有人上车的顺序。

例如： 员工数 8， 车容量 3， 员工到达顺序为 1 2 3 4 5 6 7 8， 3 个人一个小组，分三个小组， 小组一： 1， 2， 3， 小组二： 4， 5， 6， 小组三： 7， 8。 小组上车顺序为： 小组三， 小组二， 小组一 。 所有员工上车顺序为 7 8 4 5 6 1 2 3

```
package com.bean;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //小组几个人
        int MemberNum = sc.nextInt();
        //公交容量
        int BusCapacity = sc.nextInt();
        //有几人开辟多大空间，因为普通数组并不支持动态扩展
        int[] arr = new int[MemberNum];
        //依次入数组
        for (int i = 0; i < MemberNum; i++) {
            arr[i] = sc.nextInt();
        }
        //总共几个组（余 1 就多加一辆车，不能因为多一个人就省一辆车）
        int groupNum = MemberNum % BusCapacity == 0 ?
MemberNum/BusCapacity:MemberNum/BusCapacity+1;
        //从最后一组开始，递减输出
        for (int i = groupNum-1; i >= 0; i--) {
            //最后一组的车有可能坐不满，所以循环条件为小于总容量即可
            if(i == groupNum-1) {
                for (int j = i * BusCapacity;j>=0 && j < MemberNum;j++){
                    System.out.print(arr[j] + " ");
                }
                //除了最后一组其他组肯定都是满员的
            }else {
                for (int j = i * BusCapacity; j < (i+1) * BusCapacity; j++) {
                    //最后一位去掉空格（强迫症，emmm）
                    if (j == BusCapacity-1)
                        System.out.print(arr[j]+"");
                    else
                        System.out.print(arr[j]+" ");
                }
            }
        }
    }
}
```

```
    }  
}
```

10. 猿辅导公司的  $N$  位 ( $N \geq 4$ ) 研发同学组织了一次秋游活动，某同学带了个无人机在高空拍照，活动结束后，先拍了一张所有同学排成公司猴头 Logo 的照片，接着有人提议再排成“猿”的首字母 Y 字形来拍一张合照。

用字符串中的每一个字符（不是换行符或结束符'\0'）代表一位老师，输出排好后的队形。要求 Y 字除去中心点外，上下半部分等高，按照从左到右，从上到下进行排序。队形中没人的部分用空格占位。

输入数据保证可以排出一个完整的 Y 字，即长度为  $3k+1$  ( $k \geq 1$ )

例如: 7 个 x，排成队形为（为了方便说明，这里用‘-’代替空格）:

x---x

-x-x

--x

--x

--x

```
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.util.Scanner;  
public class Main{  
    public static void main(String[] args) throws IOException{  
        Scanner sc=new Scanner(System.in);  
        int num=sc.nextInt();  
        BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));  
        String str=reader.readLine();  
        char[] ch=str.toCharArray();  
        int t1=(2*num+1)/3-1,t2=(num+2)/3-1,k=0,t3=t1;  
        for(int i=0;i<=t2;i++) {  
            //打印前面的空格  
            for(int n=0;n<i;n++) System.out.print(" ");  
            //打印数字  
            for(int j=i;j<=t1;j++) {  
                if(i==j || i+j==2*t2)  
                    System.out.print(ch[k++]);  
                else System.out.print(" ");  
            }  
            System.out.println();  
            t1--;  
        }  
        //打印下面的数字  
        for(int i=t2;i<t3;i++) {  
            //打印前面的空格
```

```
        for(int j=0;j<t2;j++) {  
            System.out.print(" ");  
        }  
        System.out.println(ch[k++]);  
    }  
}
```