1、对于一个给定的正整数组成的数组 a[] ，如果将 a 倒序后数字的排列与 a 完全相同，我们称这个数组为“回文”的。
例如， [1, 2, 3, 2, 1] 的倒序是他自己，所以是一个回文的数组；而 [1, 2, 3, 1, 2] 的倒序是 [2, 1, 3, 2, 1] ，所以不是一个回文的数组。
对于任意一个正整数数组，如果我们向其中某些特定的位置插入一些正整数，那么我们总是能构造出一个回文的数组。

输入一个正整数组成的数组，要求你插入一些数字，使其变为回文的数组，且数组中所有数字的和尽可能小。输出这个插入后数组中元素的和。

例如，对于数组 [1, 2, 3, 1, 2] 我们可以插入两个 1 将其变为回文的数组 [1, 2, 1, 3, 1, 2, 1] ，这种变换方式数组的总和最小，为 11 ，所以输出为 11 。
解析：
问题可以转换为求回文子序列的最大和，则最终最优解为 2 * sum - dp[0][a.length - 1]，sum 为数组 a 所有元素的和。

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/**
 * Dynamic Programming
 *
 * State:
 *   dp[i][j]: 表示 a[i],...,a[j]中的回文子序列的最大和
 *
 * Initial State:
 *   dp[i][i] = a[i]
 *
 * State Transition:
 *   if (a[i] == a[j]) dp[i][j] = dp[i + 1][j - 1] + 2 * a[i];
 *   else dp[i][j] = max(dp[i + 1][j], dp[i][j - 1]);
 *
 * @author wylu
 */
public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        String[] strs = br.readLine().split(" ");
        int[] a = new int[n];
        long sum = 0;
        for (int i = 0; i < n; i++) {
            a[i] = Integer.parseInt(strs[i]);
            sum += a[i];
```

```
        }

        long[][] dp = new long[n][n];
        for (int i = a.length - 1; i >= 0; i--) {
            dp[i][i] = a[i];
            for (int j = i + 1; j < a.length; j++) {
                if (a[i] == a[j]) dp[i][j] = dp[i + 1][j - 1] + 2 * a[i];
                else dp[i][j] = Math.max(dp[i + 1][j], dp[i][j - 1]);
            }
        }
        System.out.println(2 * sum - dp[0][n - 1]);
    }
}
```

2、关于 java 的异常处理机制的叙述哪些正确
A.catch 部分捕捉到异常情况时，才会执行 finally 部分

B.当 try 区段的程序发生异常时，才会执行 catch 区段的程序

C.不论程序是否发生错误及捕捉到异常情况，都会执行 finally 部分

D.都不正确

答：BC

3、下列选项中,降低进程优先级的合理时机是
A.进程刚完成 I/ O,进入就绪列队

B.进程的时间片用完

C.进程长期处于就绪列队中

D.进程从就绪态转为运行态

答案:B

解析：完成 IO 进程应该提升其优先级，处于就绪队列等待调度的进程一般不会改变其优先级。

4、设有一个含有 13 个元素的 Hash 表(0~12),Hash 函数是:H(key)=key % 13,其中%是求余数运算。用线性探查法解决冲突,则对于序列(2、 8 、31、20、19、18、53、27),18 应放在第几号格中

A.5 B.9 C.4 D.0

答：B

解析：求出 18 之前的序列余数为 2、8、5、7、6，H(18)=5 与之前的冲突，直接向后移，6、7、8 都有元素，因此放在 9 号上

5、如果一个二叉树中任意节点的左右子树“高度”相差不超过 1，我们称这个二叉树为“高度平衡二叉树”。根据如上定义，一个高度为 8 的高度平衡二叉树至少有几个节点？

A.33　　B.34　　C.54　　D.55　　E.127

答：C

解析：使用递推的思想:

P(N)=N 层平衡二叉树的最少节点数

那么 P(N)=P(N-1)+P(N-2)+1

P(1)=1

P(2)=2

P(3)=P(1)+P(2)+1=4

.....

P(8)=P(6)+P(7)+1=20+33+1=54

6、Linux 文件权限一共 10 位长度，从前数第 5-7 位表示的内容是 （）

A.文件所有者的权限

B.文件所有者所在组的权限

C.其他用户的权限

D.文件类型

答案：B

解析：Linux 文件权限一共 10 位长度，分成四段 第一位表示文件类型 -表示普通文件。d 表示目录文件 第二、三、四位表示文件所有者的读，写，执行权限 第五、六、七位表示文件所在属组的读，写，执行权限 第八、九、十位表示文件的其它用户的读，写，执行权限

7、某种产品中,合格品率为 85%。一个合格品被检查成次品的概率是 10%，一个次品被检查成合格品的概率为 5%。问题：求一个被检查成合格品的产品确实为合格品的概率

A.75%　　B.85%　　C.99%　　D.91.5%

答案：C

解析：求一个被检查成合格品的产品确实为合格品的概率：分母应该为合格品检查为合格品概率和次品被检查成合格品概率，分子为合格品被检查为合格品概率。已知合格品被检查为次品概率为 10%，则合格品被检查为合格品概率为 90%；合格品率为 85%，次品率为 15%，次品被检查为合格品概率概率为 5%，则 85%*90%/（85%*90%+15%*5%）=99%.

已知一棵树的前序遍历是”GDAFEMHZ”，而中序遍历是 ”ADEFGHMZ”求后续遍历是?

DAEFHZMG B.AEFDHZMG C.ADEFGHMZ D.AFEDHMZG

答案：B

某地每天有流星雨的概率是相等的，一个人每天晚上都去观察，发现一个月能够看到流星的概率是 91%，请问半个月中能够看到流星的概率是多少

A.97% B.70% C.87% D.66.70%

答案：B

解析：一个月内看到流星雨的概率为 p1=0.91，那看不到的概率就是 p2=0.09。一个月看不到流星雨这个事件，可以拆成两个 “半个月看不到流星雨” (p3)，且相互独立。
p1=1-p2=1-(p3*p3),得到 p3=0.3。即半个月看不到流星雨的概率为 0.3,那能看到的概率自然就是 0.7。

10、在给定文件中查找与设定条件相符字符串的命令为

grep B.gzip C.find D.sort

答案：A

解析：简单来说，grep 是查找匹配条件的行，find 是搜索匹配条件的文件。此题是在指定文件中查询，所以用 grep 命令。

11、两个人抛硬币，规定第一个抛出正面的人可以吃到苹果，请问先抛的人能吃到苹果的概率多大

A.1/2 B.1/3 C.2/3 D.5/6

答案：C

解析：第一轮 1/2 正面
第一轮 1/4 双方都为反面,进入第二轮,第二轮正面概率为 1/4*1/2
类推可得:
所以总概率为 1/2+(1/2)^3+(1/2)^5+...+(1/2)^2n+1
计算后即为 2/3