

进行数据库提交操作时使用事务（Transaction）是为了？（）

正确答案: B 你的答案: 空 (错误)

提高效率

保证数据一致性

网络安全

归档数据文件

-

- 关于编译原理，以下文法中，是自顶向下分析的文法是哪几个？（）

- 正确答案: E 你的答案: 空 (错误)

- LR(0)
- SLR(1)
- LALR(1)
- LR(1)
- LL(1)

- 以下协议属于网络协议中的应用层的是（）

- 正确答案: B C 你的答案: 空 (错误)

- IP
- HTTP
- FTP
- UDP

- 以下选项属于进程间通信的是（）

- 正确答案: A B D 你的答案: 空 (错误)

- 管道
- 套接字
- 内存
- 消息队列

- 一个小朋友有 70 个玩具运往目的地，目的地距离小朋友 60 步。每次小朋友最多拿

40 个玩具，每走 2 步会掉一个玩具，则它最多能把 1 个玩具运到目的地

- 你的答案 (错误)

- 1

- 参考答案

(1) 20

- 设哈希表长 $m = 14$, 哈希函数 $H(\text{key}) = \text{key} \% 11$ 。表中已有 4 个结点: $\text{addr}(15)=4$, $\text{addr}(38)=5$, $\text{addr}(61)=6$, $\text{addr}(84)=7$, 其余地址为空。如果用二次探测再散列处理冲突, 关键字为 49 的结点的地址是 1

- 你的答案 (错误)

- 1

- 参考答案

(1) 9

- $X = +0111001, Y = +1001101$, 求 $[X - Y]$ 补 = 1

- 你的答案 (错误)

- 1

- 参考答案

(1) 11101100

-125 的反码是多少 1

你的答案 (错误)

1

参考答案

(1) 10000010

-

- 以数据集{1,6,8,2,9,4}为权值构造一棵赫夫曼树，其带权路径长度为 1

- 你的答案 (错误)

-

1

- 参考答案

(1) 70

- 组成数字 1 到 1234 的所有数字的各位的总和是 1

- 你的答案 (错误)

-

1

- 参考答案

(1) 15895

- 小李和小王各有书籍若干本,小李对小王说:“我如果给你 2 本,我们的书将一样多。”

小王说:“我如果给你 2 本,我的书籍数量将只有你的三分之一。” 请问, 小李和小

王共有书籍 1

- 你的答案 (错误)

- 1

- 参考答案

(1) 16

- 有一段楼梯台阶有 100 级台阶，以小明的脚力第 n 步能跨 n 级，请问小明登上这段楼梯需要多少步 1

- 你的答案 (错误)

- 1

- 参考答案

(1) 14

- 有 K 层的二叉树至多有 1 个节点

- 你的答案 (错误)

- 1

- 参考答案

(1) $2^k - 1$

- 1main() {
- 2 char c=040;
- 3 printf("%0\n", c<<1);
- 4}

程序的结果是 1

你的答案 (错误)

1

参考答案

(1) 100

一颗完全二叉树的节点数量为 666，那么这棵树上的叶子节点数为 1

你的答案 (错误)

1

参考答案

(1) 333

有 1 分，2 分，5 分，10 分四种硬币，每种硬币数量无限，给定 n 分钱($n < 10000$)，有多少中组合可以组成 n 分钱？

```
import java.util.Scanner;
```

```
public class Solution {  
    int count = 0;  
    int[] arr;  
  
    public int calculateWays(int n) {  
        arr = new int[n + 1];  
        return calculateWays1(n);  
    }  
    //记忆化搜索递归  
    private int calculateWays1(int n) {  
        if (n < 0)  
            throw new IllegalArgumentException("input wrong");  
        if (n == 0)  
            return 0;  
        if (n == 1)  
            return 1;  
        if (n == 2 || n == 3)  
            return 2;  
        if (n == 4 || n == 5 || n == 6 || n == 7 || n == 8 || n == 9)  
            return n - 1;  
        if (n == 10)
```

```

        return 11;
    if (arr[n] != 0)
        return arr[n];
    int res = 0;
    res = Math.max(Math.max(calculateWays1(n - 1) + 1, calculateWays1(n - 2) + 2),
Math.max(calculateWays1(n - 5) + 4, calculateWays1(n - 10) + 11));
    arr[n] = res;
    return res;
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入钱数: ");
    int n = sc.nextInt();
    Solution s = new Solution();
    int sum = s.calculateWays(n);
    System.out.println(sum);
}

```

给定 n 个柱面的高度，表示降雨某地 n 块区域的海拔高度。

计算降雨之后该地最大储水面积。如果低于地平线，也就是小于 0，则一定积水。

//单调栈

```

public int maxArea(int[] heights){
    Stack<Integer> stack = new Stack<>();
    int result = 0;
    for(int i = 0 ; i < heights.length ; i ++){
        while(!stack.isEmpty() && heights[stack.peek()] <= heights(i)){
            int curIndex = stack.pop();
            if(stack.isEmpty()) break;
            int righthIndex = i;
            int leftIndex = stack.peek();
            int d = righthIndex - leftIndex - 1;
            int h = Math.min(heights[righthIndex],heights[leftIndex]) - heights[curIndex];
            result += d*h;
        }
        stack.push(i);
    }

    return result;
}

```