第一道：两个字符串中的唯一不同字符， 典型的异或计算题，两个相同的值异或等于 0

```
 private static char solve(String input1, String input2){
        long result = 0;
        for (int i = 0; i < input1.length(); i++){
            result ^= input1.charAt(i);
        }
        for (int i = 0; i < input2.length(); i++){
            result ^= input2.charAt(i);
        }
        return (char) result;
}
```

第二道：健身房， 动态规划，主要是 Arrays.fill()的使用

```
    private static int solve(int[] trainingDays, int[] costs){
        int[] dp = new int[365];
        int minCost = 0, lastDay = 0;
        for (int trainingDay: trainingDays){
            Arrays.fill(dp, lastDay, trainingDay, minCost);
            lastDay = trainingDay;
            minCost = getMinCost(dp, costs, trainingDay);
        }
        return minCost;
    }

    private static int getMinCost(int[] dp, int[] costs, int day){
        int oneDayTicket = (day - 1 >= 0? dp[day - 1]: 0) + costs[0];
        int sevenDayTicket = (day - 7 >= 0? dp[day - 7]: 0) + costs[1];
        int thirtyDayTicket = (day - 30 >= 0? dp[day - 30]: 0) + costs[2];
        return Math.min(oneDayTicket, Math.min(sevenDayTicket, thirtyDayTicket));
    }
```

第三道：密码是否符合要求，很基础的字符串题， 主要在于输入的部分，用 nextInt()读取了用例行数之后，需要用 nextLine()换到下一行，然后再进循环，否则会把第一行的行数当做第一个数据，最后一个数据也漏了。

```
    private static boolean solve(String input) {
        if (input.length() < 8 || Character.isDigit(input.charAt(0))) {
            return false;
        }

        int[] count = new int[]{0, 0, 0};
        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);
            if (!Character.isLetterOrDigit(c)){
                return false;
```

```
                }
                if (Character.isUpperCase(c)){
                        count[0] = 1;
                } else if (Character.isLowerCase(c)){
                        count[1] = 1;
                } else {
                        count[2] = 1;
                }
        }
        return count[0] + count[1] + count[2] > 1;
}
```