

Logic-1 Memorization Problems

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    if (isSummer) {
        return 60 <= temp && temp <= 100;
    }
    else {
        return 60 <= temp && temp <= 90;
    }
}

public int sortaSum(int a, int b) {
    int sum = a + b; // get the SUM!
    if (10 <= sum && sum <= 19) {
        return 20;
    }
    else {
        return sum;
    }
}

public int teaParty(int tea, int candy) {
    if (tea < 5 || candy < 5) {
        return 0;
    }
    else if (tea >= candy * 2 || candy >= tea * 2) {
        return 2;
    }
    else {
        return 1;
    }
}

public boolean inOrder(int a, int b, int c, boolean bOk) {
    if (bOk) {
        return b < c;
    }
    else {
        return a < b && b < c;
    }
}

public boolean lastDigit(int a, int b, int c) {
    int lastA = a % 10;
    int lastB = b % 10;
    int lastC = c % 10;

    if (lastA == lastB) {
        return true;
    }
    else if (lastA == lastC) {
        return true;
    }
    else if (lastB == lastC) {
        return true;
    }
    else {
        return false;
    }
}

public boolean lastDigit(int a, int b, int c) {
    int lastA = a % 10;
    int lastB = b % 10;
    int lastC = c % 10;

    return lastA == lastB || lastA == lastC || lastB == lastC;
}

public boolean shareDigit(int a, int b) {
    int onesDigitA = a % 10;
    int onesDigitB = b % 10;
    int tensDigitA = a / 10;
    int tensDigitB = b / 10;

    return tensDigitA == tensDigitB ||
        tensDigitA == onesDigitB ||
        onesDigitA == tensDigitB ||
        onesDigitA == onesDigitB;
}
```