

## 第七章

---

### 7.34

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

public class Code7_34 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String s = input.next();
        s = sort(s);
        System.out.print("String sorted is: ");
        System.out.println(s);
        input.close();
    }

    public static String sort(String s) {
        char[] str = s.toCharArray();
        Arrays.sort(str);
        String tmp = new String(str);
        return tmp;
    }
}
```

## 第八章

---

### 8.11

```
import java.util.Scanner;

public class Code8_11 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number between 0 and 511: ");
        int num = input.nextInt();
        solve(num);
        input.close();
    }

    public static void solve(int num) {
        int[][] val = new int[3][3];
        for (int i = 2; i >= 0; i--) {
            for (int j = 2; j >= 0; j--) {
                val[i][j] = num % 2;
                num /= 2;
            }
        }
        for (int i = 0; i < val.length; i++) {
            for (int j = 0; j < val[i].length; j++) {
```

```

        if (val[i][j] == 1)
            System.out.printf("T ");
        else
            System.out.print("H ");
    }
    System.out.println("");
}
}
}

```

## 8.14

```

import java.util.Scanner;

public class Code8_14 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the size of the matrix: ");
        int num = input.nextInt();

        char[][] s = new char[num][num];
        for (int i = 0; i < num; i++) {
            String str = input.next();
            s[i] = str.toCharArray();
        }
        solve(s, num);
        input.close();
    }

    public static void solve(char[][] s, int num) {
        findrow(s, num);
        findcol(s, num);
        findmajar(s, num);
        findsub(s, num);
    }

    public static void findrow(char[][] s, int num) {
        boolean flag1 = true;
        for (int i = 0; i < num; i++) {
            boolean flag2 = true;
            for (int j = 1; j < num; j++) {
                if (s[i][j] != s[i][0]) {
                    flag2 = false;
                    break;
                }
            }
            if (flag2) {
                int row = i + 1;
                System.out.println("All " + s[i][0] + "s on row " + row);
                flag1 = false;
            }
        }
        if (flag1) {
            System.out.println("No same numbers on a row");
        }
    }
}

```

```

public static void findcol(char[][] s, int num) {
    boolean flag1 = true;
    for (int i = 0; i < num; i++) {
        boolean flag2 = true;
        for (int j = 1; j < num; j++) {
            if (s[j][i] != s[0][j]) {
                flag2 = false;
                break;
            }
        }
        if (flag2) {
            int column = i + 1;
            System.out.println("All " + s[0][i] + "s on column " + column);
            flag1 = false;
        }
    }
    if (flag1) {
        System.out.println("No same numbers on a column");
    }
}

public static void findmajar(char[][] s, int num) {
    boolean flag = true;
    for (int i = 1; i < num; i++) {
        if (s[i][i] != s[0][0]) {
            flag = false;
            break;
        }
    }
    if (flag)
        System.out.println("All " + s[0][0] + "s on the major diagonal");
    else
        System.out.println("No same numbers on the major diagonal");
}

public static void findsub(char[][] s, int num) {
    boolean flag = true;
    for (int i = 1; i < num; i++) {
        if (s[i][num - i - 1] != s[0][num - 1]) {
            flag = false;
            break;
        }
    }
    if (flag)
        System.out.println("All " + s[0][num - 1] + "s on the sub-
diagonal");
    else
        System.out.println("No same numbers on the sub-diagonal");
}
}

```

