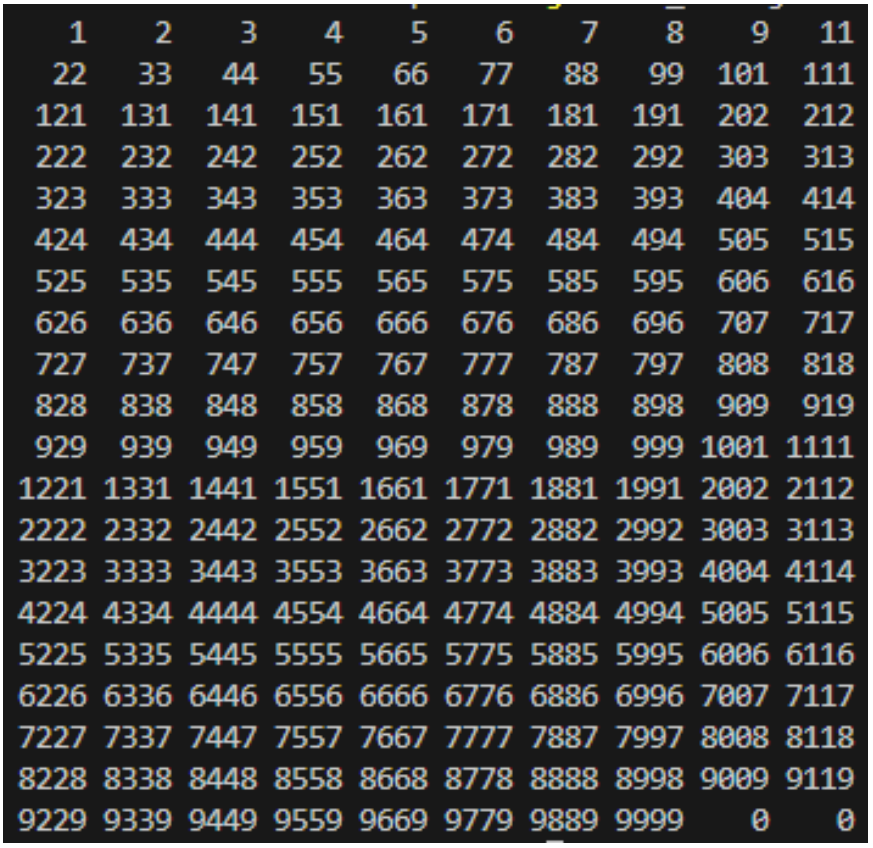
11/03 p1~p3

(1)印出 1-10000 之間有幾個迴文數字，並將它們每 10 個數字印在一列，輸出到螢幕。註：迴文數字由左到右和由右到左列都一樣，例如：121, 1221。
(2)連續輸入一連串(最多 10000 個)整數，每輸入一個數字，即輸出到目前為止的中位數。當程式讀入的數字是 99999 時結束執行。
(3)紐西蘭有一年電力供應不足，所以必積分區輪流停電，全國共分為十七個區，為了能以公平的方式輪流停電，決策者選定了一個隨機數 5，每次執行停電從第 1 區開始，接 下來是第 6 區，亦即以 5 為區的相隔進行停電，接著是第 11 區、第 16 區，因為只 有 17 區，所以從頭開始輪，已經停過的就不用再停電了。所以數了 5 次(17, 2, 3, 4, 5)， 因此輪到第 5 區停電。整個停電的順序是:
1 6 11 16 5 12 2 9 17 10 4 15 14 3 8 13 7
但是這個決策者覺得使用 5 這數字似乎不盡公平，所以讓民眾可以選擇介於2到10之間的數字 n。寫一程式可以讓使用者輸入此數字 n，並讓程式將停電的區號依序列出

(1) 迴文數字, 完種程式碼:

```
/* 11/03 楊育哲
 * 實作第一題: 1~10000 有多少迴文數字
 */
public class h1_1103 {
    public static void main(String args[]){
        int[] number_ = new int[200]; //9*2+9*10*2=18+180=198<200
        // 1, 11, 101+10, 1001+110
        for(int i=0; i<9; i++) number_[i]=i+1;
        for(int i=1; i<10; i++) number_[i+8]=i*11;
        for(int i=0; i<9; i++){
            for(int j=0; j<10; j++) number_[i*10+j+18]=(i+1)*101+j*10;
        }
        for(int i=0; i<9; i++){
            for(int j=0; j<10; j++) number_[i*10+j+108]=(i+1)*1001+j*110;
        }
        for(int i=0; i<20; i++){
            for(int j=0; j<10; j++) System.out.printf("%4d ", number_[i*10+j]);
            System.out.println("");
        }
    }
}
```

ex.



```
   1    2    3    4    5    6    7    8    9   11
  22   33   44   55   66   77   88   99  101  111
 121  131  141  151  161  171  181  191  202  212
 222  232  242  252  262  272  282  292  303  313
 323  333  343  353  363  373  383  393  404  414
 424  434  444  454  464  474  484  494  505  515
 525  535  545  555  565  575  585  595  606  616
 626  636  646  656  666  676  686  696  707  717
 727  737  747  757  767  777  787  797  808  818
 828  838  848  858  868  878  888  898  909  919
 929  939  949  959  969  979  989  999 1001 1111
1221 1331 1441 1551 1661 1771 1881 1991 2002 2112
2222 2332 2442 2552 2662 2772 2882 2992 3003 3113
3223 3333 3443 3553 3663 3773 3883 3993 4004 4114
4224 4334 4444 4554 4664 4774 4884 4994 5005 5115
5225 5335 5445 5555 5665 5775 5885 5995 6006 6116
6226 6336 6446 6556 6666 6776 6886 6996 7007 7117
7227 7337 7447 7557 7667 7777 7887 7997 8008 8118
8228 8338 8448 8558 8668 8778 8888 8998 9009 9119
9229 9339 9449 9559 9669 9779 9889 9999    0    0
```

---

(2) 求中位數，細節參考最上圖, 完種程式碼:

```
/* 11/03 楊育哲
 * 實作第二題: 輸入多串數字，每輸入一數字，即輸出到目前為止的中位數
 */
import java.util.Scanner;
public class h2_1103 {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int[] list = new int[10000];
        int value=0, top=0;
        while(true){
            value = sc.nextInt();
            if(value==99999) break;
            list[top++]=value;
            for(int i=1; i<top; i++){
                int key=list[i], j=i-1;
                while(j>=0&&key<list[j]){
                    list[j+1]=list[j];
                    j--;
                }
                list[j+1]=key;
            }
            // for(int i=0; i<top; i++) System.out.printf("%d ", list[i]);
            // System.out.printf(", %d\n", top);
            if(top%2==0) System.out.println((list[top/2-1]+list[top/2])/2);
```

```
            else System.out.println(list[top/2]);
        }
    }
}
```

ex.

test1. 說明: 依序輸入9, 99, 999, 9999, 99999

(看出有對基偶數分別處理)

9
9
99
54
999
99
9999
549
99999

test2. 說明: 依序輸入800, 200, 600, 333, 99999

(以看出有無排列後取中位數)

800
800
200
500
600
600
333
466
99999

---

(3) 擬約瑟夫問題, 詳細參考最上圖, 完整程式碼:

```
/* 11/03 楊育哲
 * 實作第三題: 擬約瑟夫問題, 17區, n=2~10
 */
import java.util.Scanner;
public class h3_1103 {
    public static void main(String args[]){
```

```
        Scanner sc = new Scanner(System.in);
        int[] list = new int[17];
        int[] listAfter = new int[17];
        int current=0, index=0;
        for(int i=0; i<17; i++) list[i] = i+1;
        System.out.print("輸入數字n(2~10), 表示每格n區停電: ");
        int n = sc.nextInt();
        while(index<17){
            listAfter[index] = list[current];
            list[current] = 0;
            int steps=n;
            while(steps>0){
                current=(current+1)%17;
                while(list[current]==0&&index<16) current=(current+1)%17;
                steps--;
            }
            index++;
        }
        for(int i=0; i<17; i++) System.out.printf("%d ", listAfter[i]);
    }
}
```

ex.

輸入數字n(2~10), 表示每格n區停電: 5
1 6 11 16 5 12 2 9 17 10 4 15 14 3 8 13 7
