

11/17 p1~p3

(1)撰寫函式 `boolean isPrime(int n)` 來判斷數入參數 `n` 是否為質數；函式 `int reverse(int n)` 則是可以將輸入數字 `n` 排列方式反轉，例如：`25 -> 52`, `1300->31`；利用這二個函式完成下列題目：若某質數其所有位數的數字反向排列仍為質數，但不同於原質數，則稱之為emirp數字。例如：13是一個質數, 31也是一個質數,因此13是一個emirp數字。除此之外，17也是一個emirp數字，因為17和71都是質數。由1開始，將找出的emirp數字由小到大排列，請問第101個emirp數字是多少？

```
/* 1117 楊育哲
 * 實作第一題：找第101個emirp
 */
public class h1_1117 {
    public static boolean isPrime(int value){
        if(value!=2&&value%2==0) return false;
        for(int i=3; i<value; i+=2){
            if(value%i==0) return false;
        }
        return true;
    }
    public static int reverse(int value){
        int ans=0;
        while(value>0){
            ans = ans*10+value%10;
            value/=10;
        }
        return ans;
    }
    public static void main(String args[]){
        int current=1, finded=1;//已加入2
        if(isPrime(2)&&isPrime(reverse(2))) System.out.printf("%d\n", 2);
        while(finded<101){
            current+=2;//3、5、7、9、...，跳過二，於前面補
            if(isPrime(current)){
                if(isPrime(reverse(current))){
                    finded++;
                    System.out.printf("%d\n", current);
                }
            }
        }
        System.out.printf("emirp 101: %d\n", current);//emirp 101
    }
}
```

| | | |
|-----|------|------|
| 2 | 389 | 1151 |
| 3 | 701 | 1153 |
| 5 | 709 | 1181 |
| 7 | 727 | 1193 |
| 11 | 733 | 1201 |
| 13 | 739 | 1213 |
| 17 | 743 | 1217 |
| 31 | 751 | 1223 |
| 37 | 757 | 1229 |
| 71 | 761 | 1231 |
| 73 | 769 | 1237 |
| 79 | 787 | 1249 |
| 97 | 797 | 1259 |
| 101 | 907 | 1279 |
| 107 | 919 | 1283 |
| 113 | 929 | 1301 |
| 131 | 937 | 1321 |
| 149 | 941 | 1381 |
| 151 | 953 | 1399 |
| 157 | 967 | 1409 |
| 167 | 971 | 1429 |
| 179 | 983 | 1439 |
| 181 | 991 | 1453 |
| 191 | 1009 | 1471 |
| 199 | 1021 | 1487 |
| 311 | 1031 | 1499 |
| 313 | 1033 | 1511 |
| 337 | 1061 | 1523 |
| 347 | 1069 | 1559 |
| 353 | 1091 | 1583 |
| 359 | 1097 | 1597 |
| 373 | 1103 | 1601 |
| 383 | 1109 | 1619 |
| | | 1657 |

(2)撰寫一函式 `int[][] twoDimSort(int[][] matrix)` 可以輸入二維陣列，將他們依照第一維度優先、第二維度次之的順序排序，例如：

輸入範例：<第一行為資料數量>

4
2 4
1 2
3 4
2 3

輸出範例

1 2
2 3
2 4
3 4

```

/* 1117 楊育哲
 * 實作第二題：二維排序
 */
import java.util.Scanner;
public class h2_1117 {
    public static void twoDimSort(int[][] array){
        for(int i=0; i<array.length-1; i++){
            int minIndex=i;
            for(int j=i+1; j<array.length; j++) if(array[minIndex][0]>array[j][0])minIndex=j;
            int[] temp = array[minIndex];
            array[minIndex] = array[i];
            array[i] = temp;
        }
        for(int i=0; i<array.length-1; i++){
            int subLen=2;
            if(array[i][0]==array[i+1][0]){
                int base = array[i][0];
                while(base==array[i+subLen][0]) subLen++;
                for(int j=i; j<i+subLen; j++){
                    int minIndex=j;
                    for(int k=j+1; k<i+subLen; k++) if(array[k][1]<array[minIndex][1])minIndex=k;
                    int[] temp = array[minIndex];
                    array[minIndex] = array[j];
                    array[j] = temp;
                }
            }
        }
    }
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int len=sc.nextInt();
        int[][] array = new int[len][2];
        for(int i=0; i<len; i++){
            array[i][0] = sc.nextInt();

```

輸入、
輸出:

4
2 4
1 2
3 4
2 3
1 2
2 3
2 4
3 4

```

        array[i][1] = sc.nextInt();
    }
    twoDimSort(array);
    for(int i=0; i<len; i++){
        for(int j=0; j<array[0].length; j++) System.out.printf("%d ", array[i][j]);
        System.out.println("");
    }
}
}

```

(3)撰寫一函式 `int[][] spinMatrix(int n, int dir)`，可以輸出一個螺旋矩陣，其中輸入參數有二個，`n` 代表要產出的螺旋矩陣為 `n*n` 大小，`20 >= n >= 1`，`dir` 代表方向，若 `dir=1` 代表順時針，`dir=2` 代表逆時針。

輸出範例：

```

1 2 3
8 9 4
7 6 5

```

輸入範例：<第一行為測試案例數量>

```

2
3 1
2 2

```

```

1 4
2 3

```

```

/* 1117 楊育哲
 * 實作第三題：旋轉矩陣
 */
import java.util.Scanner;
public class h3_1117 {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int times = sc.nextInt();
        while(--times>=0){
            int current=1, n=sc.nextInt(), dir=sc.nextInt();//dir{1:順時針/2:逆時針}
            int[][] array = new int[n][n];
            int x=0, y=0, state=0, l=0, r=n-1, oldState;
            array[0][0] = current++;
            while(current<=n*n){
                oldState = state;
                switch (state) {
                    case 0:
                        if(x==r) state=1;
                        else x++;
                        break;
                    case 1:
                        if(y==r){
                            state=2;
                            r--;
                        }else y++;
                        break;
                    case 2:
                        if(x==l){
                            state=3;
                            l++;

```

輸入、輸出：

```

3
3 1
1 2 3
8 9 4
7 6 5
2 2
1 4
2 3
4 1
1 2 3 4
12 13 14 5
11 16 15 6
10 9 8 7

```

```

        }else x--;
        break;
    case 3:
        if(y==1) state=0;
        else y--;
        break;
    }
    if(state==oldState){
        if(dir==1) array[y][x]=current++;
        else array[x][y]=current++;
    }
}
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++) System.out.printf("%2d ", array[i][j]);
    System.out.println("");
}
}
}
}

```