Contents

```
1 chapter practice
 1.1 Cramer . . . .
 2.1 BigDecimal Calculator . . . . . . . . . . .
 2.2 Magic Square
 2.3 Prime Factorization . . . . . . . . . . . .
3 homeworks
 3.1 Print Prime Numbers . . . . .
 3.2 P1 Infix Notation to Postfix Notation . . . . . . . . . . . .
 3.3 P2 Infix Notation to Postfix Notation . . . . . . . . . . . .
 4.3 The BigInteger GCD LCM and Prime . . . . . . . . . . . . . . .
 10
5 practice
 10
 5.2 Nine Nine Multiplication Table . . . . . . . . . . . . . . . .
                          11
```

1 chapter practice

1.1 Cramer

```
2 請利用克萊姆(Cramer)公式,嘗試解出2*2的線性方程式
3 \mid ax + by = e
4 | cx + dy = f
5
6|輸入:
7| 輸入a b e
8 與 c d f 分別代表兩個線性方程式的常數與答案
9 其形態皆為double
10
11|輸出:
12| 請利用 Cramer 解出 x 與 y 二 值
13 其形態 皆為 double
14 */
15
16
  import java.util.Scanner;
17
18 public class Main {
19
20
      public static void main(String[] args) {
21
          Scanner input = new Scanner(System.in);
22
23
           double x1 = 0, y1 = 0, a1 = 0, x2 = 0, y2 =
               0, a2 = 0;
24
25
          x1 = input.nextDouble();
          y1 = input.nextDouble();
26
27
          a1 = input.nextDouble();
          x2 = input.nextDouble();
28
29
          y2 = input.nextDouble();
30
          a2 = input.nextDouble();
31
32
          // 克萊姆公式
          // [a b][x] = [e]
33
           // [c \ d][y] = [f]
34
          // x = ed - bf / ad - bc
// y = af - ec / ad - bc
35
36
37
          double x = 0, y = 0;
```

1.2 calculation

```
請撰寫一個程式,讀取0-1000其中一位數,然後將其的每一位數相乘。
4
  為一介於0<= num <=1000的Integer
6
7
  形態為Integer
8
10
  import java.util.Scanner;
12
13
  public class Main {
15
      public static void main(String[] args) {
16
          Scanner input = new Scanner(System.in);
17
          String str = input.nextLine();
18
          // 先把字串內的每位數單獨拆分進 array 內
19
          String[] array = str.split("");
20
21
22
          Integer ans = new Integer(1);
23
          for( int i = 0; i < str.length(); i++ ){</pre>
24
25
              // 用 Integer.parseInt() 把 array[i]
                  內的字符轉數字
              ans = Integer.parseInt(array[i]) * ans;
26
27
28
29
          System.out.println(ans);
30
      }
31 }
```

1.3 distance

```
2| 請撰寫一個程式,提示使用者輸入兩點座標(x1,y1)及(x2,y2),並顯示區
      請注意,您可以利用Math.pow(a,0.5)來計算√a。
3
  輸入
  x1,y1: 1.5 -3.4
  x2, y2: 4 5
8|輸出
  8.764131445842194
10
11 計算兩點間距離的公式為
  (x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1) 開根號
12
13
14
  import java.util.Scanner;
15
16
  public class Main {
17
18
19
      public static void main(String[] args) {
         Scanner input = new Scanner(System.in);
20
21
          String str = input.nextLine();
22
          // 先把字串內的每位數單獨拆分進 array 內
23
24
          String[] array = str.split("");
25
26
         Integer ans = new Integer(1);
```

```
27
           for( int i = 0; i < str.length(); i++ ){</pre>
                                                             57
                                                                             a = Math.sqrt( s * (s-array[0]) *
                                                                                 (s-array[1]) * (s-array[2]) );
28
               // 用 Integer.parseInt() 把 array[i]
                                                             58
29
                   內的字符轉數字
                                                                             // 利用 BigDecimal 去取小數點並四捨五入
                                                             59
               ans = Integer.parseInt(array[i]) * ans;
                                                             60
                                                                             BigDecimal ans = new BigDecimal(a);
30
                                                                             System.out.println( ans.setScale(3,
                                                             61
31
           }
                                                                                 RoundingMode.HALF_UP));
32
                                                             62
                                                                        }
33
           System.out.println(ans);
                                                             63
34
      }
35 }
                                                             64
                                                             65
                                                                    }
                                                             66 }
```

1.4 Heron's formula

1.5 year month day

```
1 /*
2|請撰寫一程式,提示使用者輸入三角形的三個頂點座標(x1,y1)、1(/*
3 | s = (a + b + c) / 2
                                                          2 請撰寫一程式,提示使用者輸入月份與年份,接著顯示該月份的天數。
A = \sqrt{s(s - a)(s - b)(s - c)}
                                                                 舉個例子,如果使用者月份輸入2,年份輸入2012,此程式應顯示"F
                                                                2012 had 29
6| 輸入
                                                                days "。如果使用者月份輸入3,年份輸入2015,程式顯示 "March
7 三角形三個點: 1.5 -3.4 4.6 5 9.5 -3.4
                                                                 2015 had 31 days " o
8|輸出
                                                          3
                                                            */
9 面積為33.600
10
                                                          5
                                                            import java.util.Scanner;
11
  輸出格式為小數點後三位
                                                          6
                                                            public class Main {
12 | 若三點不為三角形請輸出 "Not Triangle"
                                                          8
13
                                                          9
                                                                public static void main(String[] args) {
14
                                                                    Scanner input = new Scanner(System.in);
                                                          10
15 import java.math.BigDecimal;
                                                          11
                                                                    int month = input.nextInt();
16 import java.math.RoundingMode;
                                                          12
                                                                    int year = input.nextInt();
  import java.util.Arrays;
17
                                                          13
18 import java.util.Scanner;
                                                                    // 分別存 30 天的月份 和 31 天的月份
                                                          14
19
                                                                        以及他們的英文月份名
20 public class Main {
                                                                    int[] thirty = {1,3,5,7,8,10,12};
21
                                                          15
                                                                    String[] thirty_eng = {"January", "March",
22
      private static Math Precision;
                                                          16
                                                                         "May", "July", "August", "October",
23
24
      public static void main(String[] args) {
                                                                         "December"};
25
                                                          17
          Scanner input = new Scanner(System.in);
26
                                                          18
                                                                    int[] thirty_one = {4,6,9,11};
                                                                    String[] thirty_one_eng = {"April", "June",
27
          double x1 = 0, y1 = 0, x2 = 0, y2 = 0, x3 = 0
                                                          19
                                                                         "September", "November"};
              0, y3 = 0;
                                                          20
28
29
          x1 = input.nextDouble();
                                                          21
                                                                    // 查詢他是否是 30 天的月份
30
          y1 = input.nextDouble();
                                                                    // 最後放 return 後面的其他程式碼就不會執行了
                                                          22
31
          x2 = input.nextDouble();
                                                          23
                                                                    for( int i = 0; i < thirty.length; i++){</pre>
32
          y2 = input.nextDouble();
                                                          24
                                                                        if(month == thirty[i]){
          x3 = input.nextDouble();
33
                                                                            System.out.println(thirty_eng[i] + "
                                                          25
34
          y3 = input.nextDouble();
                                                                                 " + year + " had 31 Days" );
35
                                                          26
                                                                            return;
          double[] array = new double[3];
36
                                                          27
                                                                        }
          array[0] = Math.sqrt(Math.pow(x2 - x1, 2) +
37
                                                                    }
                                                          28
              Math.pow(y2 - y1, 2);
                                                                    for( int i = 0; i < thirty_one.length; i++){</pre>
                                                          29
          array[1] = Math.sqrt(Math.pow(x3 - x2, 2) +
38
                                                                        if(month == thirty_one[i]){
                                                          30
              Math.pow(y3 - y2, 2);
                                                                            System.out.println(thirty_one_eng[i]
                                                          31
          array[2] = Math.sqrt( Math.pow(x3 - x1, 2) +
39
                                                                                + " " + year + " had 31 Days" );
              Math.pow(y3 - y1, 2));
                                                          32
                                                                            return;
40
                                                                        }
                                                          33
          // 判斷是否為三角形
41
                                                          34
          // Arrays.sort(陣列名稱) -> 自動排序
                                                          35
                                                                    if( year % 4 == 0){
42
                                                                        System.out.println("February " + year + "
          Arrays.sort(array);
                                                          36
43
                                                                            had 29 Days");
44
          if ( array[0] + array[1] <= array[2] ){</pre>
                                                          37
45
                                                                        return:
              System.out.println("Not Triangle");
                                                          38
46
                                                                    System.out.println("February " + year + " had
          }
47
                                                          39
48
          else{
                                                                        28 Days");
49
              // 海龍公式
                                                          40
                                                          41
                                                                }
50
              double s = 0;
                                                          42 }
              for( int i = 0; i < 3; i++){
51
                  s = s + array[i];
52
53
              }
54
              s = s / 2;
                                                            1.6 circle distance
55
              double a = 0;
56
```

```
binary to decimal
                                                          1.8
2 | 請撰寫一個程式,提示使用者輸入一個點座標(x,y),接著檢視該點座標是否位於中心點(0,0),半徑為10的圓內(<=10)。
3|計算距離的公式
5 (x*x + y*y) 開根號
                                                        2│請撰寫一程式,使用者輸入二進位數字,然後印出其對應的十進位數字。
                                                              輸入 0111 1001 1100 輸出 7 9 12
7|輸入
                                                        3
8 4 5
9
  9 9
                                                          import java.util.Scanner;
                                                        5
10
                                                          import java.util.StringTokenizer;
11 輸出
(4.0,5.0) is in the circle
                                                          public class Main {
                                                        8
|(9.0,9.0)| is not in the circle
14 */
                                                        10
                                                              public static void main(String[] args) {
                                                                  Scanner input = new Scanner(System.in);
15
                                                        11
16 import java.util.Scanner;
                                                        12
                                                                  String bin = input.nextLine();
17
                                                        13
18 public class Main {
                                                        14
                                                                  StringTokenizer st = new StringTokenizer(
                                                                      bin, " ");
19
      public static void main(String[] args) {
20
                                                        15
          Scanner input = new Scanner(System.in);
                                                                  while(st.hasMoreTokens()){
21
                                                        16
                                                                      int two = 1, deci = 0;
22
          double x = input.nextDouble();
                                                        17
23
          double y = input.nextDouble();
                                                        18
                                                                      String str = st.nextToken();
                                                                      for( int i = str.length()-1; i >= 0; i--
                                                        19
24
          //是否在半徑 10 以內
25
                                                                         deci += ( str.charAt(i) - '0' )*two;
          if( Math.sqrt( Math.pow(x, 2) + Math.pow(y,
                                                        20
26
                                                                         two *= 2;
              2) ) <= 10 ){
                                                        21
                                                        22
              // 如果要輸出 ("%f %d", a, b) 要用 ->
27
                                                        23
                                                                      System.out.println(deci);
                  System.out.printf
                                                                  }
                                                        24
              // %.1f 控制小數點後 1 位
28
                                                        25
              System.out.printf("(%.1f,%.1f) is in the
29
                                                        26
                                                              }
                  circle" ,x ,y);
                                                        27 }
30
          }
          else{
31
32
              System.out.printf("(\%.1f,\%.1f) is not in
                  the circle",x,y);
33
34
          System.out.println();
                                                          1.9 substring
35
      }
36 }
                                                          請撰寫一程式,使用者輸入二個字串,然後顯示二個字串是否為第一個写
                                                          輸入 string s1:ABCD string s2:BC
                                                        3
  1.7 Regular polygon area
                                                          輸出 BC is a substring of ABCD
                                                          輸入 string s1:ABCD string s2:CA
1 /*
  正多邊形為帶有n個邊,各邊等長,各角度也相同的多邊形(也就是
                                                          輸出 CA is not a substring of ABCD
  A = (n*s^2) / 4*tan(pi/n)
                                                          */
                                                        8
5|請撰寫一程式,使用者輸入邊數,以及正多邊形的邊長,接著顯稱 import java.util.Scanner;
6
                                                        12
                                                          public class Main {
7|輸入
                                                        13
8 n=邊數: 5
                                                        14
                                                              public static void main(String[] args) {
9 s=邊長: 6.5
                                                        15
                                                                  Scanner input = new Scanner(System.in);
10
                                                        16
11|輸出
                                                        17
                                                                  String str = input.nextLine();
  74.69017017488385
12
                                                        18
13 */
                                                                  // 已確認 : 切割後的空白不會存進 array
                                                        19
14
                                                        20
                                                                  String[] array = str.split(" ");
15 import java.util.Scanner;
                                                        21
16
                                                                  // 原來可以用 str.contains 去找子字串
                                                        22
  public class Main {
17
                                                                      我還以為只能找字元..
18
                                                        23
                                                                  if(array[0].contains(array[1])){
      public static void main(String[] args) {
19
                                                                     {\tt System.out.println(array[1] + " is a}
                                                        24
20
          Scanner input = new Scanner(System.in);
                                                                          substring of " + array[0]);
21
                                                        25
22
          double n = input.nextDouble();
                                                        26
                                                                  else{
          double s = input.nextDouble();
23
                                                                      System.out.println(array[1] + " is not a
24
                                                                          substring of " + array[0]);
          // java 有內建 tan 和 PI 在 Math 裡 太酷了り
25
                                                        28
                                                                  }
          System.out.println( (n*s*s) /
26
                                                        29
              (4*Math.tan(Math.PI/n));
```

}

30

31 }

27

28 }

}

54 55

56

2 exam

BigDecimal Calculator

```
57
1 /*
                                                                58
  Problem 5 4 1
                                                                59
  Write a program to input an expression string in
                                                                60
       which the operands and operators are separated by
                                                                61
       zero or more spaces. For example, 3.5*4+3 and 3.5
                                                                62
       + 4 % 3 are acceptable expressions. The operator
                                                                63
       in the expression might be +, -, *, /, and %.
                                                                64
       Your program must print out the expression and
                                                                65
       its computing result. The sample output for the
                                                                66
       input expression 3.5*4+3 is shown below:
                                                                67
4 \mid 3.5 * 4 + 3 = 17
                                                                68
                                                                69
  Requirement
                                                                70
  Write a static method BigDecimal calculate(String
       exp) to compute the expression and return a
       BigDecimal result. The operands should be stored
       as BigDecimal in this method. You have to use the
       arithmetic\ operators\ provided\ by\ the\ BigDecimal
                                                                72
       class to calculate the expression. (未依規定,以
                                                                73
       0 分計)
                                                                74
                                                                75
9
  Input
                                                                76
10 There are many input lines. Each line has an input
                                                                77
       expression Exp. There are three operands and two
                                                                78
       operators in Exp.
                                                                79
11
                                                                80
12 Output
                                                                81
13 For each input expression Exp, please output the
                                                                82
       expression and its computing result. Note that
                                                                83
       all tokens are separated by a space character.
                                                                84
       (小數點以下印一位)
                                                                85
14
                                                                86
15
                                                                87
16 import java.math.BigDecimal;
                                                                88
17 import java.math.RoundingMode;
                                                                89
18 import java.util.Arrays;
                                                                90
  import java.util.Scanner;
                                                                91
20 import java.util.StringTokenizer;
                                                                92
                                                                93
22
  public class Main {
                                                                94
23
                                                                95
24
       public static BigDecimal calculate(String exp){
                                                                96
25
                                                                97
26
           StringTokenizer st = new
                                                                98
                StringTokenizer(exp,"0123456789. ");
                                                                99
27
           StringTokenizer ma = new StringTokenizer(exp,
                                                               100
                 +-*/% ");
                                                               101
28
                                                               102
           String[] operator = new String[5];
29
                                                               103
           String[] digits = new String[5];
30
                                                               104
31
           Arrays.fill(operator, "0");
                                                               105
           Arrays.fill(digits, "0");
32
                                                               106
33
                                                               107
34
           int i = 0:
                                                               108
           boolean flag = false;
35
                                                               109
36
           while( st.hasMoreTokens() ){
                                                               110
37
                                                               111
38
                if( exp.charAt(0) == '-' && !flag ){
                                                               112
39
                    String haha = st.nextToken();
                                                               113
                    flag = true;
40
                                                               114
41
                    continue;
                                                               115
               }
42
                                                               116
43
                else{
                                                               117
                    operator[i] = st.nextToken();
44
45
                    i++;
                                                               118
               }
46
                                                               119
47
                                                               120
48
                                                               121
           int j = 0;
49
                                                               122
50
           while( ma.hasMoreTokens() ){
                                                               123
51
                digits[j] = ma.nextToken();
                                                               124
52
               j++;
```

```
}
boolean e = false;
for( int k = 0; k < j-1; k++){
    if(exp.charAt(0) == '-' ||
        operator[k].equals("0") ){
        break;
    }
    System.out.print(digits[k] + " ");
    System.out.print(operator[k] + " ");
    e = true;
if( e ){
    System.out.print(digits[j-1] + " = ");
boolean yep = false;
if( i == 2 ){
    if( operator[1].equals("*") ||
        operator[1].equals("/") ||
        operator[1].equals("%") ){
        if( operator[0].equals("+") ||
            operator[0].equals("-") ) {
            yep = true;
            String tp;
            tp = operator[1];
            operator[1] = operator[0];
            operator[0] = tp;
            String str;
            str = digits[1];
            digits[1] = digits[0];
            digits[0] = str;
            str = digits[2];
            digits[2] = digits[1];
            digits[1] = str;
        }
    }
BigDecimal sum = new BigDecimal(digits[0]);
if( exp.charAt(0) == '-'){
    BigDecimal tmp = BigDecimal.ZERO;
    sum = tmp.subtract(sum);
}
i = 0:
for( int k = 1; k < j; k++){
    BigDecimal b = new BigDecimal(digits[k]);
    if( operator[i].equals("+") ){
        sum = sum.add(b);
    else if( operator[i].equals("-") ){
        if(vep){
            sum = b.subtract(sum);
        }
        else{
            sum = sum.subtract(b);
    }
    else if( operator[i].equals("*") ){
        sum = sum.multiply(b);
    else if( operator[i].equals("/") ){
        sum = sum.divide(b, 1
            RoundingMode.CEILING);
    else if( operator[i].equals("%") ){
        sum = sum.remainder(b);
    }
    i++;
}
```

```
125
           return sum;
       }
126
       public static void main(String[] args) {
127
           Scanner input = new Scanner(System.in);
128
129
130
           while( input.hasNext() ){
                String exp = input.nextLine();
131
132
                BigDecimal ans = calculate(exp);
                // BigDecimal 四捨五入的寫法 .setScale(
133
                    想要留取的位數, RoundingMode.HALF_UP)
                System.out.println(ans.setScale(1,
134
                    RoundingMode.HALF_UP));
135
           }
       }
136
137 }
```

2.2 Magic Square

```
1 /*
2 Problem
3 If you have good observations skills, you may found
       that building a Magic Square is simple. A Magic
       Square has only an odd number N of rows and
       columns where N < 100. A Magic Square is created
       by integer numbers in the range from 1 to N2 ,
       with a peculiar property, the "sum of the
       numbers" in each row, column and diagonal are
       the same.
  For the case n = 3.
  M. Square
                                     Columns
       Diagonals
  4 9 2
                      4+9+2 = 15
                                     4+3+8 = 15
       2+5+8 = 15
8 3 5
                      3+5+7 = 15
                                     9+5+1 = 15
       4+5+6 = 15
9 8 1 6
                      8+1+6 = 15
                                     2+7+6 = 15
10
11 Input
12
  Each line contains an Integer N denoting an N * N
       Magic Square.
13
14 Output
15 如果 N 是偶數則輸出 " It is not an odd number. "。
       如果 N 是奇數則先輸出直橫列的加總數字,再輸出 N *
       N 數字陣列,每個數字以%5d 格
       式輸出。每組測資間請空一行。
16 */
17
18 import java.util.Arrays;
19 import java.util.Scanner;
20
  public class Main {
21
22
       public static void main(String[] args) {
23
          Scanner input = new Scanner(System.in);
24
25
          boolean flag = false;
26
27
           while(input.hasNext()){
               int n = input.nextInt();
28
29
               if( flag ){
30
                   System.out.println();
31
32
               if( n % 2 == 0 ){
33
34
                   System.out.println("It is not an odd
                       number.");
35
                   flag = true;
36
                   continue;
              }
37
38
39
               int[][] square = new int[100+5][100+5];
               for( int i = 0; i < n; i++ ){</pre>
40
41
                   Arrays.fill( square[i], 0);
42
```

```
43
                 int r = n-1, c = n/2;
44
45
                 square[r][c] = 1;
                 for( int i = 2; i <= n*n; i++ ){</pre>
46
47
48
                     int rn = r+1;
49
                     int cn = c+1;
50
                     if( rn >=n ){
51
                          rn = 0;
52
53
                     if(cn >= n){
                          cn = 0;
54
55
                     }
                     if( square[rn][cn] != 0 ){
56
57
                          rn = r-1;
                          cn = c;
58
59
                          if( rn < 0 ){
60
                              rn = n-1;
61
62
                     }
63
                     square[rn][cn] = i;
64
                     r = rn;
65
                     c = cn;
66
67
                 int sum = 0;
68
                for( int i = 0; i < n; i++ ){</pre>
                     sum += square[0][i];
69
70
                }
71
                 System.out.println(sum);
72
                 for( int i = 0; i < n; i++ ){</pre>
                     for( int j = 0; j < n; j++ ){</pre>
73
74
                          System.out.printf("%5d",
                               square[i][j]);
75
76
                     System.out.println();
77
78
                 flag = true;
            }
79
80
81
       }
82 }
```

2.3 Prime Factorization

1 /*

5

10

12

13

16

```
2 Problem
3|輸入數字 N (資料型態為 Integer),
     請輸出該數字的所有質因數及其次方。例如 N=360=23
     *32 *5。此題數字可能會有質數出現。
  Requirement
  請撰寫以下兩個 static methods: (未依規定,以 Ø 分計)
  1. boolean [] PrimeArray(long N) {....}
     which returns an array A of Boolean values, where
         A[i] is true if i is a prime number,
         otherwise, A[i] is false if i is not a prime
         number. Note that A.length = N+1; Hint: if n
         is a prime number, then n * j is not a prime,
         where j \ge 2;
11
  2. String PrimeFactorization(long N) {…}
     which returns a string of prime factorization for
         the number N. For example, if N = 360, the
         returned string is "2^3 * 3^2 * 5"
  Input
14
  輸入有多列,每列有個整數 N,最多 1000 列。
15
17 Output
18 | 第一行輸出所有數字中之最大數 X及其開根號整數 X, 其後針對每一組測資
     輸出 N的質因數分解,將數字
     N的所有質因數(及其次方)以小到大方式顯示出來,如質因
     數之次方數大於 1,以 / 運算符號顯示,不同質因數間以
```

```
* 運算符號互相連接, *運算符號前 後加空格。
                                                              94
19
                                                                         // 可以用 method 回傳陣列
                                                              95
20
                                                                              但也要用陣列接他的回傳值
21 import java.util.Arrays;
                                                                         boolean[] A = PrimeArray(max);
                                                              96
22
  import java.util.Scanner;
                                                              97
                                                                         System.out.println(max + " " +
23
                                                                             (int)Math.sqrt(max) );
24 public class Main {
                                                                         for( int o = 0; o < k; o++ ){</pre>
                                                              98
25
                                                              99
26
      public static boolean [] PrimeArray(long N){
                                                             100
                                                                             if( A[N[o]] ){
27
                                                             101
                                                                                  System.out.println(N[o]);
28
           boolean[] prime = new boolean[1000000];
                                                                             }
                                                             102
           Arrays.fill(prime, true);
29
                                                             103
                                                                             else{
30
                                                             104
                                                                                  String factorization =
           for( long i = 2; i <= N; i++ ){</pre>
31
                                                                                      PrimeFactorization(N[o]);
32
               if(prime[(int)i]){
                                                             105
                                                                                  System.out.println(factorization);
                   for( long j = i * i; j <= N; j += i ){</pre>
33
                                                             106
                                                                             }
                       prime[(int)j] = false;
34
                                                             107
                                                                         }
35
                                                                     }
                                                             108
36
               }
                                                             109 }
37
           }
38
           return prime;
39
      public static String PrimeFactorization(long N){
40
                                                                     homeworks
41
42
           int[] num = new int[1000000];
           Arrays.fill( num, 0 );
43
                                                                      Print Prime Numbers
                                                                3.1
44
45
           long count = N;
46
           for( int i = 2; i < N; i++ ){</pre>
47
               while( count % i == 0 ){
                                                                Write a program to input an integer N and print all
                   num[i]++;
48
                                                                     prime numbers less than N.
49
                   count /= i;
                                                                Please print 10 prime numbers in each line.
50
               }
51
               if( count == 1 ){
                                                               5
52
                   break;
                                                                import java.util.Scanner;
                                                               6
53
               }
                                                               7
                                                                import java.util.Vector;
54
           }
                                                               8
55
                                                                public class Main {
           // 因數轉字串處理
56
                                                              10
57
           String[] tp = new String[1000000];
                                                              11
                                                                     public static void main(String[] args) {
58
           int j = 0;
                                                              12
                                                                         Scanner input = new Scanner(System.in);
           for( int i = 2; i < N; i++ ){</pre>
                                                                         int n = input.nextInt();
59
                                                              13
60
               if( num[i] != 0 ){
                                                              14
                   tp[j++] = Integer.toString(i);
61
                                                              15
                                                                         // 建質數表
62
                   if( num[i] != 1 ){
                                                                         // java 中沒有全域變數
                                                              16
                       tp[j++] = "
63
                                                                             所以這邊建表目前是每次輸入都重建一次
64
                        tp[j++] =
                                                                              但不夠漂亮
                            Integer.toString(num[i]);
                                                              17
                                                                         boolean[] prime = new boolean[10000000+5];
65
                                                              18
                   tp[j++] = " * ";
66
                                                              19
                                                                         // 用 vector 存質數們
               }
67
                                                                         Vector<Integer> v = new Vector<Integer>();
                                                              20
           }
68
                                                                         for( int i = 2; i <= n; i++ ){</pre>
                                                              21
69
                                                              22
                                                                             if(!prime[i]){
           // 合併字串 用 str1.concat(str2)
70
                                                              23
                                                                                 v.add(i);
           // 因為最後會多一組 * 所以j-1
71
                                                              24
                                                                                  for( int j = i * i; j <= n; j += i ){</pre>
72
           String result = "";
                                                              25
                                                                                      prime[j] = true;
73
           for( int i = 0; i < j-1; i++){
                                                              26
               result = result.concat(tp[i]);
74
                                                                             }
                                                              27
75
           }
                                                              28
                                                                         }
76
           return result;
                                                              29
77
                                                                         for( int i = 0, k = 0; i < v.size(); i++, k++</pre>
                                                              30
78
79
      public static void main(String[] args) {
                                                              31
           Scanner input = new Scanner(System.in);
80
                                                              32
                                                                             if( k == 9 || i == v.size() - 1){
           int max = 0;
81
                                                              33
                                                                                  System.out.println(v.get(i));
           int[] N = new int[1000+5];
82
                                                                                  // 把 k 歸回去 -1 是因為下次輪迴會先
                                                              34
           int k = 0;
83
                                                                                      k++
84
                                                              35
                                                                                  k = -1;
85
           while( input.hasNextLong() ) {
                                                                             }
                                                              36
86
                                                              37
                                                                             else{
87
               N[k] = input.nextInt();
                                                                                  System.out.print(v.get(i) + " ");
                                                              38
88
               if (N[k] > max) {
                                                              39
                   max = N[k];
89
                                                              40
90
               }
                                                              41
                                                                         v.clear();
91
               k++;
                                                              42
                                                                     }
           }
92
                                                              43 }
```

93

```
3.2 P1 Infix Notation to Postfix Notation
                                                     61
                                                               System.out.println();
                                                     62
                                                     63
                                                               // BigDecimal 四捨五入的寫法 .setScale(
1 /*
                                                                   想要留取的位數, RoundingMode.HALF_UP)
2 為了要便於中序轉後序的程式運行,請先將算式的數字與運算符號
                                                               System.out.println(ans.setScale(3,
3 1. 使用 String 類別 & 字串輸入
                                                                   RoundingMode.HALF_UP));
4 2. 使用 String Tokenizer 類別 或
                                                     65
                                                           }
      String類別之split()method來切割中序運算式。
                                                     66 }
5|3. 使用String[] array 儲存所有tokens。
6 | 4. 將數字 token轉換成實數,加總後輸出總和。 Hint: Use
      BigDecimal in java.math package
                                                        3.3 P2 Infix Notation to Postfix Notation
7
8 輸入
  每組測資會給予不一定長度的算式,符號包含+,-,*,/,(,),%,算量/*
9
                                                      2 請利用 P1
10
                                                            中序轉後序-分割的答案輸出後續的算式,並輸出後序運算式
11|輸出
12 | 第一行輸出運算式的每個運算元,運算元間以空格字元隔開。
                                                      4 程式要求
13 第二行輸出運算式的每個運算子,運算子間以空格字元隔開。
                                                      5 1.必須使用 java.util.Stack 實作,否則不予計分。
  第三行印出運算元的總和,小事點以下印三位。
14
                                                        2. 請利用part1求得的token 字串陣列轉後序
15
16
17 import java.math.BigDecimal;
                                                        每組測資會給予不一定長度的算式,符號包含+,-,*,/,(,),%,算式中數
18 import java.math.RoundingMode;
                                                      9
19 import java.util.Scanner;
                                                     10
20
                                                        輸出
                                                     11
21 public class Main {
                                                        將後序算式的每個運算元與運算子後加入一個空格字元印出。
                                                     12
22
                                                        格式請詳看sample output
                                                     13
      public static void main(String[] args) {
23
         Scanner input = new Scanner(System.in);
24
                                                        import java.util.Scanner;
                                                     15
25
         String str = input.nextLine();
                                                        import java.util.Stack;
                                                     16
         String token = "+-*/()%";
26
                                                     17
27
                                                        public class Main {
                                                     18
         // 當字串中帶有 + * / \ 等符號時要寫 \\
28
                                                      19
             轉義,因為他們在正則表達示中有相應的不同意義20
                                                            public static int icpfunc(String check){
         // limit is 0; array contains all substrings
29
                                                               // icp 是該 token 的順序
                                                     21
         // Positive Lookahead or Lookbehind 觀念
30
                                                               // icp -> +- 1 -> */% 2 -> ( 4
                                                     22
             "((?=@)|(?<=@))"
                             -> 才會把 運算子 還有
31
                                                     23
                                                               int icp;
                                                     24
                                                               if(check.equals("(")){
             運算元 都單獨切開
                                                     25
                                                                   icp = 4;
32
         String[] array =
             str.split("((?=\\+/-/\\*//\\)/\\(/%)/(?<=\\26)
                                                                   return icp;
33
                                                     27
                                                               else if(check.equals("*") ||
         // 預設 ans 賦值 可控制小數點位數 ex 0.000 ->
                                                     28
34
                                                                   check.equals("/") || check.equals("%") ){
             小數點後三位
                                                     29
                                                                   icp = 2;
         BigDecimal ans = new BigDecimal("0.000");
35
                                                     30
                                                                   return icp;
36
                                                               }
                                                     31
         boolean flag = true;
37
                                                               else{
                                                     32
38
         for( String check : array){
                                                     33
                                                                   icp = 1;
             // string.contains
39
                                                     34
                                                                   return icp;
                 一次查詢多個關鍵字的寫法
                                                     35
                 要另外存一個數組 token
                                                      36
             // 且在查詢時要以此 for ( String check :
40
                                                            public static int ispfunc(String check){
                                                     37
                 array ) 去做掃描
                                                               // isp 是堆疊最上層 stack.peek 中的順序
                                                     38
41
             if( !token.contains(check) ){
                                                      39
                                                               // isp -> ( 0 -> +- 1 -> */% 2
                 // 將字串型態的數字轉成實數並相加
42
                                                               int isp;
                                                     40
                BigDecimal b = new BigDecimal(check);
43
                                                      41
                                                               if(check.equals("(")){
                 // (boolean 判斷式)? (true的輸出):
44
                                                      42
                                                                   isp = 0;
                    (false 的輸出)
                                                     43
                                                                   return isp;
45
                 System.out.print( flag ? check : " "
                                                     44
                    + check);
                                                               else if(check.equals("*") ||
                                                     45
46
                 flag = false;
                                                                   check.equals("/") || check.equals("%") ){
47
                                                     46
                                                                   isp = 2;
                                                                   return isp;
                 // BigDecimal 內建 .add 可以直接相加
                                                      47
48
                                                               }
                                                     48
                ans = ans.add(b);
49
                                                     49
                                                               else{
50
             }
                                                                   isp = 1;
                                                      50
51
52
         System.out.println();
                                                     51
                                                                   return isp;
                                                     52
                                                               }
53
54
         flag = true;
                                                     53
                                                           }
                                                     54
55
         for( String again : array){
                                                      55
                                                            public static void main(String[] args) {
             if( token.contains(again) ){
56
                                                               Scanner input = new Scanner(System.in);
                                                     56
57
                 System.out.print( flag ? again :
                                                     57
                                                               String str = input.nextLine();
                    + again);
                                                     58
                                                               String token = "+-*/(%";
58
                 flag = false;
                                                     59
59
             }
         }
                                                     60
                                                               // 用 split 切割字串
60
```

8

```
5 每組測資為一整數 N,請注意本題輸入數字的數目不定。
61
          String[] array =
              str.split("((?=\\+/-/\\*//\\)/\\(/%)/(?<=\\+6
                                                           Output Format
62
           // 使用 stack 儲存運算子
                                                          8|請輸出前 N 個 Palindromic Prime(由 2
63
          Stack<String> stack = new Stack<>();
64
                                                                開始)。輸出的方式為每行顯示 10
          int icp = 0, isp = 0;
65
                                                                個數字,每行數字間以空格分開,但最後一個數字不用空格直接換行
66
           for (String s : array) {
67
                                                         10
                                                           Technical Specification
              // 如果是 ) 則輸出所有 ( 以前的 stack
68
                                                           • 0 \le N \le 100
                                                         11
                   內的運算子
                                                           */
                                                         12
              // 但 () 都不輸出
                                                         13
69
                                                           import java.util.Arrays;
70
              if (s.equals(")")) {
                                                         14
                  while (!stack.peek().equals("(")) {
                                                         15
                                                           import java.util.Scanner;
71
72
                      System.out.print(stack.peek() + "
                                                         16
                                                           import java.util.Vector;
                                                         17
                                                         18
                                                           public class Main {
73
                      stack.pop();
                  }
                                                         19
74
                                                                public static void main(String[] args) {
                                                         20
75
                  // 把 ) 也 pop 掉
                                                         21
76
                  stack.pop();
                                                                    // 直接在最開始就建好質數表
              }
                                                         22
77
                                                         23
                                                                   boolean[] prime = new boolean[100000+5];
              // 如果是 + - * / %
78
                                                         24
                                                                   Arrays.fill(prime, false);
              else if (token.contains(s)) {
79
                                                         25
                                                                   Vector<String> v = new Vector<String>();
                  // 運算此運算子的 icp 和 當前 stack
80
                                                         26
                      頂端的運算子的 isp
                                                                   for( int i = 2; i <= 100000; i++ ){
                                                         27
                  icp = icpfunc(s);
81
                                                         28
82
                  if (stack.empty()) {
                                                         29
                                                                       if(!prime[i]){
                      stack.push(s);
83
                                                                           // 因為 100000 * 100000 會超出 int
                                                         30
84
                  }
                                                                               所以用 long
                  else {
85
                                                                           // i * i 是 int -> 強制轉型 (long)
                                                         31
                      isp = ispfunc(stack.peek());
86
                                                         32
                                                                           for(long j = (long) i * i; j <=</pre>
                      if (icp > isp) {
87
                                                                               100000; j += i ) {
88
                          stack.push(s);
                                                                               // prime 的指標是放 int
                                                         33
89
                      } else {
                                                         34
                                                                               prime[(int) j] = true;
                          while (icp <=
90
                                                                           }
                                                         35
                              ispfunc(stack.peek())) {
                              {\tt System.out.print(stack.peek()}^{6}
91
                                  + " ");
                                                                           // 先把 i 轉成 String (
                                                         37
                              stack.pop();
92
                                                                               因為這樣比較方便 reverse )
                              if (stack.empty()) {
93
                                                         38
                                                                           String str = String.valueOf(i);
94
                                  break;
                                                         39
95
                                                                           // 使用 StringBuilder 宣告 rev ->
                                                         40
96
                                                                               因為 StringBuilder 才有
97
                          stack.push(s);
                                                                               .reverse() 可以用
                      }
98
                                                                           StringBuilder rev = new
                                                         41
99
                  }
                                                                               StringBuilder();
              }
100
                                                                           // 這裡 append (附加) str 的值給 rev
                                                         42
              // 如果是運算元則直接輸出
101
                                                         43
                                                                           rev.append(str);
102
                                                         44
                                                                           // reverse
                  System.out.print(s + " ");
103
                                                         45
                                                                           rev.reverse();
104
                                                         46
          }
105
                                                                           // 兩字串判斷相等在 java 要用 .equals
                                                         47
           // 最後因為有可能 array
106
                                                                               不能直接 == (原因跟記憶體位址有關
               在切割的時候最後一個值是運算元
                                                                           // 然後因為 rev 是 StringBuilder 但
                                                         48
           // 那樣就還會有運算子在 stack 還沒輸出
107
                                                                               .equals 是 String 的 所以要
              所以這裡在確認一次
                                                                               .toString()
108
          while(!stack.empty()){
                                                                           if( str.equals(rev.toString()) ){
                                                         49
              System.out.print(stack.peek() + " ");
109
                                                         50
                                                                               v.add(str);
              stack.pop();
110
                                                                           }
                                                         51
111
                                                         52
112
          System.out.println();
                                                         53
                                                                       }
113
                                                                   }
                                                         54
114 }
                                                         55
                                                         56
                                                                   Scanner input = new Scanner(System.in);
                                                         57
                                                         58
                                                                   // 重複輸入寫法
       old exams
                                                         59
                                                                   while( input.hasNextInt() ){
                                                                       int n = input.nextInt();
                                                         60
   4.1 PA Palindromic Prime
                                                         61
                                                                       if( n == 0 ){
                                                         62
                                                         63
                                                                           System.out.println();
 1 /*
                                                         64
                                                                           continue;
   Palindromic Prime
                                                                       }
                                                         65
       指數字為一質數並且迴文,請撰寫一個找出迴文質數的程式66
       Ex: 757 與 313 為迴文質數
                                                         67
                                                                       for( int i = 0, k = 0; i < n; i++, k++){
                                                                           if( k == 9 || i == n - 1 ){
```

4 Input Format

```
FJCU
                                                    ENDL\N
                                                                                                           9
69
                     System.out.println(v.get(i));
                                                      47
                                                            }
                     k = -1;
                                                      48 }
70
71
                 }
                 else{
72
73
                     System.out.print(v.get(i) + " ");
                                                              The BigInteger GCD LCM and Prime
74
75
             }
76
         }
                                                       1
77
      }
                                                       2
                                                         請利用BigInteger解題,不然不予給分
78 }
                                                       3
                                                        Problem
                                                       4
                                                       5 請輸入兩個數字M、N,找出M、N的最大公因數(GCD -
  4.2 You can say 11
                                                             Greatest Common Divisor) G及最小公倍數(LCM -
                                                             Least Common
                                                             Multiple)L,並找出大於L的前5個可能質數(Prime)。
2 給你一個正整數 N,判定它是否是 11 的倍數。提示:
                                                       7
      在這裡,我們需要用到一個關於 11 倍
                                                         輸入有多筆測資,每列代表一組測資,每組測資會有兩個數字M、N
3 數的小常識,那就是:11
                                                             (數值可能會超過1ong可儲存的範圍)
      倍數的「奇數位數字和」與「偶數位數字和」兩者的差必定為
      11 的
                                                      10 Output
4|倍數。
                                                         第一列先輸出兩數的最大公因數
                                                             G及最小公倍數L,第二列再輸出大於L的前E個可能質數,每個質數
6 Input Format
                                                      12
7 每列資料有一個正整數 N,N 最大可能到 1000 位數。若 N
                                                      13
      = 0 代表輸入結束
                                                      14
                                                         import java.math.BigInteger;
                                                         import java.util.Scanner;
                                                      15
9 Output Format
                                                      16
10 先輸出奇數的和與偶數位的和,再對每一個輸入的數,輸出是否為
                                                         public class Main {
                                                      18
      11 的倍數。輸出格式請
                                                      19
                                                            public static void main(String[] args) {
11
                                                                Scanner input = new Scanner(System.in);
                                                      20
  本題必須使用 char[] or String 儲存數入之正整數。
12
                                                      21
                                                                BigInteger M, N, G, L;
13
                                                      22
14
                                                      23
                                                                while( input.hasNextBigInteger() ) {
15 import java.util.Scanner;
                                                      24
16
                                                                    M = input.nextBigInteger();
                                                      25
17
  public class Main {
                                                      26
                                                                    N = input.nextBigInteger();
18
                                                      27
      public static void main(String[] args) {
19
                                                                    // BigInteger 內建函數 gcd
                                                      28
20
         Scanner input = new Scanner(System.in);
                                                      29
                                                                    G = M.gcd(N);
21
22
         while( input.hasNext() ){
                                                      30
                                                                    // L = M * N / G
                                                                    L = M.multiply(N).divide(G);
                                                      31
23
             String str = input.nextLine();
                                                                    System.out.print(G + " " + L + "\n");
             if( str.equals("0") ){
                                                      32
24
                                                      33
25
                 break:
             }
                                                      34
                                                                    for( int i = 0; i < 5; i++){
26
                                                      35
27
                                                                        // .nextProbablePrime ->
                                                      36
28
             int even = 0, odd = 0;
                                                                            返回一個大於該 BigInteger 的質數
29
             for( int i = str.length()-1; i >= 0; i -=
                 2){
                                                      37
                                                                        System.out.print( i == 4 ?
                                                                           L.nextProbablePrime() + "\n" :
                 // 要抓字串中的每個 char 要
30
                                                                           L.nextProbablePrime() + " ");
                     str.charAt()
                 // 要把 char 轉 int 蠻複雜的
                                                      38
                                                                        L = L.nextProbablePrime();
31
                                                                    }
                                                      39
                 // 要先 String.valueOf() 然後
32
                                                      40
                                                                }
                     Integer.parseInt
                                                            }
                                                      41
                 odd += Integer.parseInt(
33
                     String.valueOf( str.charAt(i) ) );
34
             for( int i = str.length()-2; i >= 0; i -=
35
                 2){
                                                         4.4 Latin Square
                 even += Integer.parseInt(
36
                     String.valueOf( str.charAt(i) ) );
             }
37
                                                         Description
38
39
             int sub = Math.abs( even - odd );
                                                         A Latin square is an n-by-n array filled with n
             if( sub % 11 != 0){
                                                             different Latin letters, each occurring exactly
40
41
                 System.out.println( odd + " " + even
                                                             once in each row and once in each column.
                     + "\n" + str + " is not a
                                                         Write a program that prompts the user to enter the
                     multiple of 11.");
                                                             number n and the array of characters, as shown in
42
             }
                                                             the sample output, and checks if the input array
```

6 Input

is a Latin square. The characters are the first n

Enter number n, and enter n rows of letters separated

characters starting from A.

by spaces.

else{

}

}

11.");

System.out.println(odd + " " + even

+ "\n" + str + " is a multiple of

43

44

45

46

```
8 \mid 0 < n <= 26, all the letters in square are A ~ Z.
                                                              83
                                                                              }
9 Input is terminated by a set where n = -1. This set
                                                              84
       should not be processed.
                                                              85
                                                                              for( int i = 0; i < n; i++ ){</pre>
                                                                                  for( int j = 0; j < n; j++ ){</pre>
10
                                                              86
11 Output
                                                                                      // char 的讀取寫法
                                                              87
12 Recognize if it is Latin Square or not. There are 2
                                                                                           .next().charAt(0) -> 這個 0
       different kinds of your answers:
                                                                                           是固定的
13 The input array is a Latin square
                                                                                           因為單字元的索引值就是 0
14 The input array is not a Latin square
                                                              88
                                                                                      square[i][j] =
15
  Overall, just check if it is Latin square or not.
                                                                                           input.next().charAt(0);
16
                                                                                       // 計算每位是否接是出現 n 次 ->
                                                              89
17 | Sample Input
                                                                                           每行都出現一次
18 4
                                                                                      count[ square[i][j] - 'A' ]++;
                                                              90
19 A B C D
                                                                                  }
                                                              91
20 B A D C
                                                              92
                                                                              }
21 C D B A
                                                              93
                                                                              // 吃最後的換行
22 D C A B
                                                              94
                                                                              String space = input.nextLine();
23
                                                              95
24 3
                                                              96
                                                                              boolean flag = false;
25 B C A
26 B A C
                                                              97
                                                                              for( int i = 0; i < n; i++ ) {</pre>
                                                              98
                                                                                  for (int j = i; j < n; j++) {
27 C B A
                                                              99
28
                                                                                      if (row[i][square[i][j] - 'A'] ||
                                                              100
29 3
                                                                                           row[j][square[j][i] - 'A'] ||
30 B A C
                                                                                           col[i][square[j][i] - 'A'] ||
31 A C B
                                                                                           col[j][square[i][j] - 'A']) {
32 C B A
                                                              101
                                                                                           flag = true;
33
34 4
                                                             102
                                                                                           break;
                                                              103
35 D C B A
                                                              104
                                                                                      row[i][square[i][j] - 'A'] = true;
36 J A V A
                                                                                      row[j][square[j][i] - 'A'] = true;
                                                              105
37 U C C U
                                                                                      col[i][square[j][i] - 'A'] = true;
                                                              106
38 P P A P
                                                                                      col[j][square[i][j] - 'A'] = true;
                                                             107
39
                                                              108
40
  - 1
41
                                                             109
                                                                                  }
                                                                                  if (flag) {
                                                             110
42 Sample Output
                                                             111
                                                                                      break;
  The input array is a Latin square
43
                                                                                  }
                                                             112
44
                                                             113
45
  The input array is not a Latin square
                                                                              for( int i = 0; i < 26; i++ ){</pre>
                                                             114
46
                                                                                  if( count[i] != n && count[i] != 0 ){
                                                             115
47 The input array is a Latin square
                                                             116
                                                                                      flag = true;
48
                                                             117
                                                                                      break:
  The input array is not a Latin square
49
                                                                                  }
                                                             118
50
  */
                                                             119
                                                                              }
51
                                                              120
52 import java.util.Arrays;
                                                             121
                                                                              if(flag){
53 import java.util.Scanner;
                                                                                  System.out.println("The input array
                                                              122
54
                                                                                       is not a Latin square");
  public class Main {
55
                                                                              }
                                                             123
56
                                                              124
       public static void main(String[] args) {
57
                                                                                  System.out.println("The input array
                                                             125
58
           Scanner input = new Scanner(System.in);
                                                                                       is a Latin square");
59
           boolean newline = false;
                                                              126
60
                                                              127
                                                                              newline = true;
           while( input.hasNextInt() ){
61
                                                              128
                                                                         }
62
                                                                     }
                                                             129
63
               int n = input.nextInt();
                                                              130 }
64
               if( n < 0 ){
                   break:
65
               }
66
67
               // 控制最後不要換行
                                                                 5
                                                                      practice
               if( newline ){
68
69
                   System.out.println();
70
               }
71
                                                                       star pyramid
72
               int[] count = new int[26+5];
               char[][] square = new char[26+5][26+5];
73
               boolean[][] row = new boolean[26+5][26+5];
74
                                                                 3
                                                               2
               boolean[][] col = new boolean[26+5][26+5];
75
                                                               3
76
                                                                  ***
               // 陣列初始化
77
                                                                 ****
78
               for( int i = 0; i < 26; i++ ){
                                                                 */
                   Arrays.fill(count, 0);
79
                   Arrays.fill(square[i], '0');
80
                                                               8
                                                                 import java.util.Scanner;
                   Arrays.fill(row[i], false);
81
82
                   Arrays.fill(col[i], false);
```

10 public class Main {

```
11
       public static void main(String[] args) {
12
           Scanner input = new Scanner(System.in);
13
14
           int n = input.nextInt();
15
16
            for( int i = 1; i <= n; i++ ){</pre>
17
18
                for( int j = n-i; j > 0; j-- ){
                    System.out.printf(" ");
19
20
21
                for( int j = 0; j < i; j++){
22
23
                    System.out.printf("*");
24
25
                for( int j = 1; j < i; j++ ){</pre>
                    System.out.printf("*");
26
27
28
29
                System.out.println();
30
           }
31
32
       }
33 }
```

5.2 Nine Nine Multiplication Table

```
1 /*
  請列出所有九九乘法表。
2
3
  example:
  1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 ...
       1 * 9 = 9 \setminus n
5 *每個公式中間都有一個空格
6 | *每 行 最 後 都 沒 有 空 格 , 但 有 換 行 符 號
7 | *每個公式的答案都是 %2d <-重點
8 *可以利用System.out.printf(String , value ,.....);
9
  */
10
11 public class Main {
12
      public static void main(String[] args) {
13
14
           for( int i = 1; i < 10; i++ ){</pre>
15
16
               for( int j = 1; j < 10; j++){
17
                   if( j == 9 ){
18
                       System.out.printf( "%d * %d =
19
                           %2d\n'', i, j, i * j );
20
                       continue:
21
                   }
22
23
                   System.out.printf( "%d * %d = %2d ",
                       i, j, i * j );
24
25
              }
          }
26
27
      }
28
29 }
```

5.3 prime number

```
1 /*
2 請輸入一個 Integer並輸出小於其的所有質數
3 每行答案中間都有空格
4 每行最後都為換行符號(沒有空格)
5 */
6 import java.util.*;
8 public class Main {
10 public static void main(String[] args) {
```

```
13
           Scanner input = new Scanner(System.in);
           int n = input.nextInt();
14
15
           boolean prime[] = new boolean[1000];
16
17
           // java 的 vector 寫法似乎是如此 :D
18
           // <> 內不能寫 int 要寫 Integer
19
20
           Vector<Integer> v = new Vector();
21
           // Arrays.fill 類似 C++ 的 memset(a, false,
22
               sizeof(a) );
           Arrays.fill( prime, false );
23
24
           for( int i = 2; i < n; i++ ){</pre>
25
26
               if( !prime[i] ){
27
28
                   for( int j = i * i; j < n; j += i ){</pre>
29
                        prime[j] = true;
30
                    // 像 C++ 的 v.push_back(i)
31
32
                   v.add(i);
33
34
               }
35
           }
36
37
           int i;
38
           for( i = 0; i < v.size()-1; i++ ){</pre>
39
40
               // 不能直接寫 v[i] 要寫 v.get(i)
               System.out.printf( "%d ", v.get(i) );
41
42
43
44
           System.out.printf( "%d\n", v.get(i) );
45
46
      }
47 }
```

5.4 factor

```
1 /*
2 | 你的任務是,給你一個正整數 N,判定它是否是 11 的倍數。
3
4
  Input
5 每列資料有一個正整數N,N 最大可能到 1000 位數。
6 若 N = 0 代表輸入結束。
7
8
  Output
  對每個輸入的數,輸出是否為 11 的倍數。輸出格式請參考
9
      Sample Output o
10
11
12
  import java.util.Scanner;
13
14
  public class Main {
15
16
      public static void main(String[] args) {
17
18
         Scanner input = new Scanner(System.in);
19
         // java 中要用 .hasNext() 來判斷是否輸入結束
20
21
         while( input.hasNext() ){
22
             // String 的 input 是 .nextLine()
23
            String str = input.nextLine();
24
25
             // java 用 ==
26
                進行比較的時候,比較的是他們在記憶體中的存放地場
             // 而 String, Integer, Date 這些類中 equals
27
                可以用來做比較
             if( str.equals("0") ){
28
29
                break;
30
31
             boolean flag = false;
32
```

```
33
               int first = 0, second = 0;
34
               for( int i = 0; i < str.length(); i++ ){</pre>
35
                   if( flag ){
36
37
                        first += Integer.parseInt(
                            String.valueOf( str.charAt(i)
                            ));
                        flag = false;
38
                   }
39
40
                   else{
                        second += Integer.parseInt(
41
                            String.valueOf( str.charAt(i)
                            ));
                        flag = true;
42
43
                   }
               }
44
45
               // java 的 abs 前面要加 Math.
46
               if( Math.abs( first - second ) % 11 == 0
47
                   System.out.printf("%s is a multiple
48
                        of 11.\n", str );
49
               }
               else{
50
                   System.out.printf("%s is not a
51
                        multiple of 11.\n", str );
52
               }
53
           }
54
55
       }
56 }
```