Contents

1 Graph

1.1 C129

```
#include <bits/stdc++.h>

using namespace std;
char oil[100][100] = {0};
int m, n;

void dfs( int i, int j )
{
    oil[i][j] = '*';
    if( oil[i-1][j-1] == '@' )
    {
        if( i-1 >= 0 && j-1 >= 0 )
        {
            oil[i-1][j-1] = '*';
            dfs( i-1, j-1 );
        }
    }
    else if( oil[i-1][j] == '@' )
    {
        if( i-1 >= 0 )
        {
            oil[i-1][j] = '*';
            dfs( i-1, j );
        }
    }
    else if( oil[i-1][j+1] == '@' )
    {
        if( i-1 >= 0 && j+1 <= n )
        {
            oil[i-1][j+1] = '*';
            dfs( i-1, j+1 );
        }
    }
    else if( oil[i][j-1] == '@' )
    {
        if( j-1 >= 0 )
        {
            oil[i][j-1] = '*';
            dfs( i, j-1 );
        }
    }
    else if( oil[i][j+1] == '@' )
    {
        if( j+1 <= n )
        {
            oil[i][j+1] = '*';
            dfs( i, j+1 );
```
```
        }
    }
    else if( oil[i+1][j-1] == '@' )
    {
        if( i+1 <= m && j-1 >= 0 )
        {
            oil[i+1][j-1] = '*';
            dfs( i+1, j-1 );
        }
    }
    else if( oil[i+1][j] == '@' )
    {
        if( i+1 <= m )
        {
            oil[i+1][j] = '*';
            dfs( i+1, j );
        }
    }
    else if( oil[i+1][j+1] == '@' )
    {
        if( i+1 <= m && j+1 <= n )
        {
            oil[i+1][j+1] = '*';
            dfs( i+1, j+1 );
        }
    }
}

int main(void)
{
    while( cin >> m >> n )
    {
        int ans = 0;
        if(( m == 0 ) && ( n == 0 ))
        {
            break;
        }
        else
        {
            for( int i = 0 ; i < m ; i++ )
            {
                for(int j = 0 ; j < n ; j++ )
                {
                    cin >> oil[i][j];
                }
            }
        }
        for( int i = 0 ; i < m ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
            {
                if( oil[i][j] == '@' )
                {
                    dfs( i, j);
                    ans++;
                }
            }
        }
        cout << ans <<endl;
    }
    return 0;
}
```

1.2 11935

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int num, flag = 1;
    cin >> num;
    while( num > 0 )
    {
        int n, ans = 0;
```

```
12        char map[100][100] = {0};
13        cin >> n;
14        for( int i = 0 ; i < n ; i++ )
15        {
16            for(int j = 0 ; j < n ; j++ )
17            {
18                cin >> map[i][j];
19            }
20        }
21        for( int i = 0 ; i < n ; i++ )
22        {
23            for(int j = 0 ; j < n ; j++ )
24            {
25                if( map[i][j] == 'x' )
26                {
27                    ans++;
28                }
29            }
30        }
31        cout << "Case " << flag << ": " << ans <<endl;
32        num--;
33        flag++;
34    }
35    return 0;
36 }
```

2 Numbers

2.1 CongruenceEquation

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  long long Mode(long long a, long long n, long long m)
6  {
7      long long sum = 1;
8      for( ; n ; n >>= 1 )
9      {
10         if( n & 1 )
11         {
12             sum = ( sum * a ) % m;
13         }
14         a = ( a * a ) % m;
15     }
16     return sum;
17 }
18
19 int main(void)
20 {
21     int a, b, p, x, ans = 0;
22     cin >> a >> b >> p >> x;
23     for( int i = 1 ; i < x + 1 ; i++ )
24     {
25         int n;
26         n = i % p;
27         n = n * Mode( a, i, p);
28         if( n % p == b % p )
29         {
30             ans++;
31         }
32     }
33     cout << ans <<endl;
34     return 0;
35 }
```

3 JAVApractice

3.1 practice1

```
1  package com.company;
2  import java.util.Scanner;
3  public class Main {
4
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7          int n = scanner.nextInt();
8          int m = n-1;
9          for( int i = 1 ; i <= 2*n-1 ; i=i+2 ) {
10             for( int j = m ; j > 0 ; j-- ) {
11                 System.out.print(" ");
12             }
13             m--;
14             for (int t = 0; t < i; t++) {
15                 System.out.print("*");
16             }
17             System.out.println();
18         }
19     }
20 }
21 }
```

3.2 emirp

```
1  import java.lang.Math;
2  public class Main {
3
4      private static boolean prime ( int number){
5          for (int i = 2; i <= Math.sqrt(number); i++)
6          {
7              if (number % i == 0)
8              {
9                  return false;
10             }
11         }
12         return true;
13     }
14     private static boolean palindrome ( int number){
15         String numstr = number + "";
16         int left = 0;
17         int right = numstr.length() - 1;
18         while (left < right) {
19             if (numstr.charAt(left) !=
                    numstr.charAt(right)) {
20                 return true;
21             }
22             left++;
23             right--;
24         }
25         return false;
26     }
27     public static int reverse(int num){
28         int tot = 0, buf = 1;
29         boolean jud = true;
30         while (num > 0) {
31             tot = tot * buf + (num % 10);
32             num /= 10;
33             if(jud) buf *= 10;
34             jud = false;
35         }
36         return tot;
37     }
38     public static void main(String[] args) {
39         int flag = 0, num = 120;
40         for (int i = 2; i < 100000; i++) {
41             if (palindrome(i) && flag < num) {
42                 if (prime(i) && prime(reverse(i))) {
43                     System.out.print(i);
44                     flag++;
45                     if (flag % 10 == 0) {
46                         System.out.println();
47                     }
48
49                     if (flag % 10 != 0 && flag !=
                            num) {
50                         System.out.print(" ");
```

```
51                    }
52
53                    if (flag == num && flag % 10 !=
                          0) {
54                        System.out.println();
55                    }
56                }
57            }
58        }
59    }
60 }
```

3.3 practice3

```
1  package com.company;
2  import java.util.Scanner;
3  public class Main {
4
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7          int n = scanner.nextInt();
8          if( n >= 2 ){
9              System.out.print(2);
10         }
11         for (int j = 3; j < n ; j++)
12         {
13             boolean answer = true;
14             for (int i = 2; i <= Math.sqrt(j) ; i++)
15             {
16                 if (j % i == 0)
17                 {
18                     answer = false;
19                     break;
20                 }
21             }
22             if (answer)
23             {
24                 System.out.print( " "+ j );
25
26             }
27         }
28         System.out.println();
29     }
30 }
```

3.4 practice4

```
1  package com.company;
2  import java.util.Scanner;
3  public class Main {
4      private static String str;
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7          while( scanner.hasNext() ) {
8              str = scanner.next();
9              if(str.equals("0")){
10                 break;
11             }
12             int tot1 = 0, tot2 = 0;
13             for (int i = 0; i < str.length(); i += 2)
                   {
14                 tot1 = tot1 + str.charAt(i) - '0';
15             }
16             for (int j = 1; j < str.length(); j += 2)
                   {
17                 tot2 = tot2 + str.charAt(j) - '0';
18             }
19             if( tot1 > tot2 ){
20                 judgment( tot1, tot2 );
21             }
22             else{
23                 judgment( tot2, tot1 );
24             }
```

```
25             }
26         }
27
28         public static void judgment( int a, int b ){
29             int judge = a - b;
30             if( judge % 11 == 0 ){
31                 System.out.println( str +" is a multiple
                      of 11." );
32             }
33             else{
34                 System.out.println( str +" is not a
                      multiple of 11." );
35             }
36         }
37 }
```

3.5 HW1

```
1  package com.company;
2  import java.math.BigDecimal;
3  import java.util.Scanner;
4  import java.util.StringTokenizer;
5
6  public class Main {
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          String str = scanner.next();
10         BigDecimal ans = new BigDecimal(0);
11         String[] num = new String[50];
12         String[] sign = new String[50];
13
14         int flag = 0, flagg = 0;
15
16         StringTokenizer token = new
                StringTokenizer(str,"+-*/%,()",true);
17         while(token.hasMoreTokens()){
18             String str1 = token.nextToken();
19             if( Character.isDigit(str1.charAt(0))){
20                 num[flag] = str1;
21                 if( flag > 0 ){
22                     System.out.print(" ");
23                 }
24                 System.out.print(num[flag]);
25                 flag++;
26             }
27             else{
28                 sign[flagg] = str1;
29                 flagg++;
30             }
31         }
32         System.out.println();
33
34         for(int i = 0 ; i < sign.length ; i++ ){
35             if(sign[i] == null){
36                 break;
37             }
38             else if(i > 0){
39                 System.out.print(" ");
40             }
41             System.out.print(sign[i]);
42         }
43         System.out.println();
44
45         for( int i = 0 ; i < num.length ; i++ ){
46             if( num[i] == null ){
47                 break;
48             }
49             BigDecimal cal = new BigDecimal(num[i]);
50             ans = ans.add(cal);
51         }
52         System.out.printf("%.3f",ans);
53         System.out.println();
54     }
55 }
```

3.6　primenumber

```java
package com.company;
import java.lang.Math;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int num = scanner.nextInt();
        int[] arr = new int[1000];
        int flag = 0;
        for (int j = 2; j < num ; j++)
        {
            boolean answer = true;
            for (int i = 2; i <= Math.sqrt(j); i++)
            {
                if (j % i == 0)
                {
                    answer = false;
                    break;
                }
            }
            if (answer)
            {
                arr[flag] = j;
                flag++;
            }
        }
        for(int i = 0 ; i < flag ; i++){
            int temp = i+1;
            System.out.print(arr[i]);
            if( temp % 10 != 0 && i != flag -1){
                System.out.print(" ");
            }

            if( i == flag -1 && temp % 10 != 0){
                System.out.println();
            }

            if( temp % 10 == 0){
                System.out.println();
            }
        }
    }
}
```

3.7　palindromeprime

```java
package com.company;
import java.util.Scanner;
import java.lang.Math;
public class Main {

    private static boolean prime ( int number){
        for (int i = 2; i <= Math.sqrt(number); i++)
        {
            if (number % i == 0)
            {
                return false;
            }
        }
        return true;
    }
    private static boolean palindrome ( int number){
        String numstr = number + "";
        int left = 0;
        int right = numstr.length() - 1;
        while (left < right) {
            if (numstr.charAt(left) !=
                numstr.charAt(right)) {
                return false;
            }
            left++;
            right--;
```

```java
        }
        return true;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNext()) {
            int num = scanner.nextInt();
            int flag = 0;
            for (int i = 2; i < 100000; i++) {
                if(num == 0){
                    System.out.println();
                    break;
                }

                if (palindrome(i) && flag < num) {
                    if (prime(i)) {
                        System.out.print(i);
                        flag++;
                        if (flag % 10 == 0) {
                            System.out.println();
                        }

                        if (flag % 10 != 0 && flag !=
                            num) {
                            System.out.print(" ");
                        }

                        if (flag == num  && flag % 10
                            != 0) {
                            System.out.println();
                        }
                    }
                }
            }
        }
    }
}
```

3.8　magicsquare

```java
package com.company;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while(scanner.hasNext()){
            int n = scanner.nextInt();
            if (n % 2 == 0){
                System.out.println("It is not an odd
                    number.");
                if(scanner.hasNext()){
                    System.out.println();
                }
                continue;
            }
            int sum = (n * ((n * n) + 1))/2;
            System.out.println(sum);

            long[][] square = new long[n][n];
            int row = n-1;
            int col = n/2;
            square[row][col] = 1;
            for (long i = 2 ; i <= n*n ; i++) {
                if (square[(row + 1) % n][(col + 1) %
                    n] == 0) {
                    row = (row + 1) % n;
                    col = (col + 1) % n;
                }
                else {
                    row = (row - 1 + n) % n;
                }
                square[row][col] = i;
            }

            for (int i = 0 ; i < n ; i++) {
```

```
35              for (int j = 0 ; j < n ; j++) {
36                  System.out.printf("%5d",
                        square[i][j]);
37              }
38              System.out.println();
39          }
40          if(scanner.hasNext()){
41              System.out.println();
42          }
43      }
44  }
45 }
```

```
60          System.out.printf("%d\n",
                (int)Math.sqrt(maxn));
61          boolean[] A = PrimeArray(maxn);
62          for(int i = 0 ; i < flag ; i++){
63              if(A[max[i]]){
64                  System.out.println(max[i]);
65              }
66              else{
67                  System.out.println(PrimeFactorization(max[i]));
68              }
69          }
70      }
71 }
```

3.9 primefactorization

```
1  import java.util.Scanner;
2
3  public class Main {
4      private static boolean[] PrimeArray(long N){
5          boolean[] A = new boolean[(int)N+1];
6          A[0] = true;
7          A[1] = true;
8          for(long j = 2; j <= N ; j++) {
9              boolean judge = true;
10             for (int i = 2; i <= Math.sqrt(j); i++) {
11                 if (j % i == 0) {
12                     judge = false;
13                     A[(int) j] = false;
14                     break;
15                 }
16             }
17             if(judge){
18                 A[(int)j] = true;
19             }
20         }
21         return A;
22     }
23     private static String PrimeFactorization(long N){
24         String str = "";
25         boolean jud = false;
26         for(long i = 2 ; N > 1 ; i++) {
27             int flag = 1;
28             if(N % i == 0) {
29                 if(jud) {
30                     str = str + " * ";
31                 }
32                 N = N / i;
33                 str = str.concat(Long.toString(i));
34                 jud = true;
35                 while (N % i == 0) {
36                     N = N / i;
37                     flag++;
38                 }
39                 if(flag > 1){
40                     str = str + "^";
41                     str =
                        str.concat(Long.toString(flag));
42                 }
43             }
44         }
45         return str;
46     }
47     public static void main(String[] args) {
48         Scanner scanner = new Scanner(System.in);
49         int maxn = 0, flag = 0;
50         int[] max = new int[1000];
51         while(scanner.hasNextInt()){
52             int N = scanner.nextInt();
53             max[flag] = N;
54             if(max[flag] > maxn){
55                 maxn = max[flag];
56             }
57             flag++;
58         }
59         System.out.print(maxn + " ");
```

3.10 calendar

```
1  import java.util.Scanner;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7          String[] month =
                {"January","February","March","April","May","June",
8          int year = scanner.nextInt();
9          int d = scanner.nextInt();
10         int t = 0;
11         boolean leap = (year % 4 == 0 && year % 100
                != 0) || year % 400 == 0;
12         for(int i = 0 ; i < 12 ; i++) {
13             System.out.println("           " +
                    month[i] + " " + year);
14             System.out.print("----------------------------\n
                    Sun Mon Tue Wed Thu Fri Sat \n");
15             if( d == 7){
16                 d = 0;
17             }
18             t = 0;
19             for(int j = 0 ; j < d ; j++){
20                 System.out.print("    ");
21                 t++;
22             }
23             int day, nd = 1;
24             switch(i+1){case 4: case 6: case 9: case
                    11: day = 30;break;
25                 case 2: if(leap){day = 29;}else{day =
                        28;}break;
26                 default: day = 31;break;}
27             while(nd <= day){
28                 if(t % 7 == 0 && t != 0) {
29                     System.out.println();
30                     t = 0;
31                 }
32                 System.out.printf("%4d", nd);
33                 nd++;
34                 t++;
35             }
36             System.out.println();
37             d = t;
38             if(i != 11)
39                 System.out.println();
40         }
41     }
42 }
```

3.11 latinsquare

```
1  import java.util.Scanner;
2  public class Main {
3
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int num, white = 0;
```

```
 7            if(( num= scanner.nextInt()) > 0 )
 8            {
 9                int[] latin = new int[26] ;
10                String[][] square = new String[num][num];
11                String[] eng
                      ={"A","B","C","D","E","F","G","H","I","J",...
12                int flag = 0;
13                boolean ha = true;
14                for(int i = 0 ; i < num ; i++)
15                {
16                    int flagg = 0;
17                    while( scanner.hasNext() && flagg <
                          num)
18                    {
19                        square[i][flagg] = scanner.next();
20                        if((square[i][flagg].charAt(0) -
                              '0') >
                              (eng[num].charAt(0)-'0')){
21                            System.out.printf("Wrong
                                  input: the letters must
                                  be from %s to %s\n",
                                  eng[0], eng[num-1]);
22                            ha = false;
23                            break;
24                        }
25                        flagg++;
26                    }
27                    if(flagg != num && ha){
28                        System.out.printf("Wrong input:
                              you need to enter exactly %d
                              letters\n", num);
29                        ha = false;
30                        break;
31                    }
32                    flag++;
33                }
34                if(flag != num && ha){
35                    System.out.printf("Wrong input: you
                          need to enter exactly %d
                          letters\n", num);
36                    ha = false;
37                }
38                if(ha) {
39                    int rl = 0;
40                    for (int i = 0; i < num; i++) {
41                        int[] row = new int[26];
42                        for (int j = 0; j < num; j++) {
43                            for (int t = 0; t < 26; t++) {
44                                if
                                    (square[i][j].equals(eng[t
                                    {
45                                    row[t]++;
46                                    latin[t]++;
47                                }
48                            }
49                        }
50                        for (int s = 0; s < 26; s++) {
51                            if (row[s] > 1) {
52                                rl++;
53                            }
54                        }
55                    }
56                    int cl = 0;
57                    for (int i = 0; i < num; i++) {
58                        int[] column = new int[26];
59                        for (int j = 0; j < num; j++) {
60                            for (int t = 0; t < 26; t++) {
61                                if
                                    (square[j][i].equals(eng[t
                                    {
62                                    column[t]++;
63                                }
64                            }
65                            for (int s = 0; s < 26; s++) {
66                                if (column[s] > 1) {
67                                    cl++;
68                                }
```

```
69                        }
70                    }
71                }
72                int lc = 0;
73                for (int i = 0; i < 26; i++) {
74                    if (latin[i] > 0) {            ,"Y","Z"]
75                        lc++;
76                    }
77                }
78
79                if (white != 0) {
80                    System.out.println("");
81                }
82
83                if (rl == 0 && cl == 0 && lc == num) {
84                    System.out.println("The input
                          array is a Latin square");
85                } else {
86                    System.out.println("The input
                          array is not a Latin square");
87                }
88                white++;
89            }
90        }
91    }
92 }
```

3.12 consectiveFour

```
 1 import java.util.Scanner;
 2 public class Main {
 3     private static int n;
 4     private static int m;
 5     public static void main(String[] args) {
 6         Scanner scanner = new Scanner(System.in);
 7         n = scanner.nextInt();
 8         m = scanner.nextInt();
 9         int[][] array = new int[n][m];
10         for(int i = 0 ; i < n ; i++)
11         {
12             for(int j = 0 ; j < m ; j++)
13             {
14                 array[i][j] = scanner.nextInt();
15             }
16         }
17         System.out.println(isConsectiveFour(array));
18     }
19     public static boolean isConsectiveFour(int[][]
           values){
20         for(int i = 0 ; i < n ; i++)
21         {
22             for(int j = 0 ; j < m ; j++)
23             {
24                 int count = 0;
25                 if((j+3 < m) && (values[i][j] ==
                       values[i][j+3])){
26                     for(int k = j+1 ; k < j+4 ; k++){
27                         if(values[i][k] ==
                               values[i][j])
28                             count++;
29                     }
30                     if(count == 3)
31                         return true;
32                     else
33                         count = 0;
34                 }
35                 if((i+3 < n) && (values[i][j] ==
                       values[i+3][j])){
36                     for(int k = i+1 ; k < i+4 ; k++){
37                         if(values[k][j] ==
                               values[i][j])
38                             count++;
39                     }
40                     if(count == 3)
41                         return true;
42                     else
```

```
43                    count = 0;
44                }
45                if((i+3 < n) && (j+3 < m) &&
                      (values[i][j] ==
                      values[i+3][j+3])){
46                    for(int k = 1 ; k < 4 ; k++){
47                        if(values[i+k][j+k] ==
                                values[i][j])
48                            count++;
49                    }
50                    if(count == 3)
51                        return true;
52                    else
53                        count = 0;
54                }
55                if((i+3 < n) && (j-3 >= 0) &&
                      (values[i][j] ==
                      values[i+3][j-3])){
56                    for(int k = 1 ; k < 4 ; k++){
57                        if(values[i+k][j-k] ==
                                values[i][j])
58                            count++;
59                    }
60                    if(count == 3)
61                        return true;
62                    else
63                        count = 0;
64                }
65            }
66        }
67        return false;
68    }
69 }
```

3.13   rationalnumber

```
1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          while (scanner.hasNextLine()) {
7              String op, num1, num2;
8              op = scanner.nextLine();
9              char co = op.charAt(0);
10             num1 = scanner.nextLine();
11             num2 = scanner.nextLine();
12             Rational n1 = new Rational(M(num1),
                   D(num1));
13             Rational n2 = new Rational(M(num2),
                   D(num2));
14             switch (co) {
15                 case '+' : {
16                     System.out.println(n1.outRationl()
                           + " + " + n2.outRationl() + "
                           = " +
                           (n1.add(n2)).outRationl());
17                     break;
18                 }
19                 case '-' : {
20                     System.out.println(n1.outRationl()
                           + " - " + n2.outRationl() + "
                           = " +
                           (n1.sub(n2)).outRationl());
21                     break;
22                 }
23                 case '*' : {
24                     System.out.println(n1.outRationl()
                           + " * " + n2.outRationl() + "
                           = " +
                           (n1.multiply(n2)).outRationl());
25                     break;
26                 }
27                 case '/' : {
28                     System.out.println(n1.outRationl()
                           + " / " + n2.outRationl() + "
                           = " +
                           (n1.divide(n2)).outRationl());
29                     break;
30                 }
31                 default : {
32                 }
33             }
34         }
35     }
36     public static int M(String str) {
37         int s, e, molecular;
38         s = str.indexOf('(');
39         e = str.indexOf('/');
40         molecular = Integer.parseInt((str.substring(s
               + 1, e)).trim());
41         return molecular;
42     }
43     public static int D(String str) {
44         int s, e, denominator;
45         s = str.indexOf('/');
46         e = str.indexOf(')');
47         denominator =
               Integer.parseInt((str.substring(s + 1,
               e)).trim());
48         return denominator;
49     }
50 }
51 class Rational {
52     private int molecular, denominator;
53     public static int gcd(int x, int y) {
54         int Max, min, temp;
55         Max = Math.max(x, y);
56         min = Math.min(x, y);
57         if(min != 0)
58             temp = Max % min;
59         else
60             temp = Max;
61         while (temp != 0) {
62             Max = min;
63             min = temp;
64             temp = Max % min;
65         }
66         return Math.abs(min);
67     }
68     public static int lcm(int x, int y) {
69         return (x * y / gcd(x, y));
70     }
71     public Rational() {
72         molecular = 1;
73         denominator = 1;
74     }
75     public Rational(int im, int id) {
76         molecular = im;
77         denominator = id;
78     }
79     public int getMolecular() {
80         return molecular;
81     }
82     public int getDenominator() {
83         return denominator;
84     }
85     public void setMolecular(int im) {
86         this.molecular = im;
87     }
88     public void setDenominator(int id) {
89         this.denominator = id;
90     }
91     public Rational add(Rational p) {
92         int gtemp, ltemp, mtemp;
93         Rational r = new Rational();
94         ltemp = lcm(this.denominator, p.denominator);
95         mtemp = this.molecular * (ltemp /
               this.denominator) + p.molecular * (ltemp
               / p.denominator);
96         gtemp = gcd(mtemp, ltemp);
97         r.molecular = mtemp / gtemp;
98         r.denominator = ltemp / gtemp;
```

```
 99          return r;
100      }
101      public Rational sub(Rational p) {
102          int gtemp, ltemp, mtemp;
103          Rational r = new Rational();
104          ltemp = lcm(this.denominator, p.denominator);
105          mtemp = this.molecular * (ltemp /
                  this.denominator) - p.molecular * (ltemp
                  / p.denominator);
106          gtemp = gcd(mtemp, ltemp);
107          r.molecular = mtemp / gtemp;
108          r.denominator = ltemp / gtemp;
109          return r;
110      }
111      public Rational multiply(Rational p) {
112          Rational r = new Rational();
113          int temp;
114          r.molecular = this.molecular * p.molecular;
115          r.denominator = this.denominator *
                  p.denominator;
116          temp = gcd(r.molecular, r.denominator);
117          r.molecular = r.molecular / temp;
118          r.denominator = r.denominator / temp;
119          return r;
120      }
121      public Rational divide(Rational p) {
122          Rational r = new Rational();
123          int temp;
124          r.molecular = this.molecular * p.denominator;
125          r.denominator = this.denominator *
                  p.molecular;
126          temp = gcd(r.molecular, r.denominator);
127          r.molecular = r.molecular / temp;
128          r.denominator = r.denominator / temp;
129          return r;
130      }
131      public String outRationl() {
132          int temp = gcd(molecular, denominator);
133          if ((molecular % denominator) == 0) {
134              return (String.valueOf(molecular /
                      denominator));
135          }
136          else {
137              if (denominator < 0) {
138                  return "(" + (molecular / temp * -1)
                          + "/ " + (denominator / temp *
                          -1) + ")";
139              }
140              else {
141                  return "(" + (molecular / temp) + "/
                          " + (denominator / temp) + ")";
142              }
143          }
144      }
145 }
```

3.14   token

```
 1 import java.util.Scanner;
 2 import java.util.StringTokenizer;
 3
 4 public class Main {
 5     public static void main(String[] args) {
 6         Scanner scanner = new Scanner(System.in);
 7         while(scanner.hasNextLine()) {
 8             String str = scanner.nextLine();
 9             str = str.replaceAll(" ", "");
10             str = str.replaceAll("[+]","#+#");
11             str = str.replaceAll("[-]","#-#");
12             str = str.replaceAll("[*]","#*#");
13             str = str.replaceAll("[/]","#/#");
14             StringTokenizer st = new
                    StringTokenizer(str, "#");
15             String[] arr = new String[3];
16             int i = 0;
17             while (st.hasMoreTokens()) {
```

```
18                 String str1 = st.nextToken();
19                 arr[i] = str1;
20                 i++;
21             }
22             switch(arr[1]){
23                 case"+":
24                     int add =
                            Integer.parseInt(arr[0]) +
                            Integer.parseInt(arr[2]);
25                     System.out.println(arr[0] + " + "
                            +arr[2] + " = " + add);
26                     break;
27                 case"-":
28                     int sub =
                            Integer.parseInt(arr[0]) -
                            Integer.parseInt(arr[2]);
29                     System.out.println(arr[0] + " - "
                            +arr[2] + " = " + sub);
30                     break;
31                 case"*":
32                     int mul =
                            Integer.parseInt(arr[0]) *
                            Integer.parseInt(arr[2]);
33                     System.out.println(arr[0] + " * "
                            +arr[2] + " = " + mul);
34                     break;
35                 case"/":
36                     int div =
                            Integer.parseInt(arr[0]) /
                            Integer.parseInt(arr[2]);
37                     System.out.println(arr[0] + " / "
                            +arr[2] + " = " + div);
38                     break;
39             }
40         }
41     }
42 }
```

3.15   throw

```
 1 import java.util.Scanner;
 2
 3 public class Main {
 4     public static void main(String[] args) {
 5         Scanner scanner = new Scanner(System.in);
 6         while(scanner.hasNextLine()){
 7             String str = scanner.nextLine();
 8             try{
 9                 Fraction fraction = new Fraction(str);
10                 System.out.println(fraction.turn(str));
11             }
12             catch(NumberFormatException e){
13                 System.out.println("String no
                        Binary");
14                 System.exit(1);
15             }
16         }
17     }
18 }
19 class NumberFormatException extends Exception{}
20 class Fraction{
21     public Fraction(String str) throws
            NumberFormatException{
22         for(int i = 0 ; i < str.length() ; i++){
23             if(str.charAt(i) != '0' && str.charAt(i)
                    != '1'){
24                 throw new NumberFormatException();
25             }
26         }
27     }
28     public int turn(String str){
29         return Integer.parseInt(str, 2);
30     }
31 }
```