

Contents

1	Graph	1
1.1	C129	1
1.2	11935	1
2	Numbers	2
2.1	CongruenceEquation	2
3	JAVAppractice	2
3.1	practice1	2
3.2	emirp	2
3.3	practice3	3
3.4	practice4	3
3.5	HW1	3
3.6	primenumber	4
3.7	palindromeprime	4
3.8	magicsquare	4
3.9	primefactorization	5
3.10	calendar	5
3.11	latinsquare	5
3.12	consecutiveFour	6
3.13	rationalnumber	7
3.14	token	8
3.15	throw	8
3.16	Point	9

1 Graph

1.1 C129

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4 char oil[100][100] = {0};
5 int m, n;
6
7 void dfs( int i, int j )
8 {
9     oil[i][j] = '*';
10    if( oil[i-1][j-1] == '@' )
11    {
12        if( i-1 >= 0 && j-1 >= 0 )
13        {
14            oil[i-1][j-1] = '*';
15            dfs( i-1, j-1 );
16        }
17    }
18    else if( oil[i-1][j] == '@' )
19    {
20        if( i-1 >= 0 )
21        {
22            oil[i-1][j] = '*';
23            dfs( i-1, j );
24        }
25    }
26    else if( oil[i-1][j+1] == '@' )
27    {
28        if( i-1 >= 0 && j+1 <= n )
29        {
30            oil[i-1][j+1] = '*';
31            dfs( i-1, j+1 );
32        }
33    }
34    else if( oil[i][j-1] == '@' )
35    {
36        if( j-1 >= 0 )
37        {
38            oil[i][j-1] = '*';
39            dfs( i, j-1 );
40        }
41    }
42    else if( oil[i][j+1] == '@' )
43    {
44        if( j+1 <= n )
45        {
46            oil[i][j+1] = '*';

```

```

47        dfs( i, j+1 );
48    }
49 }
50 else if( oil[i+1][j-1] == '@' )
51 {
52     if( i+1 <= m && j-1 >= 0 )
53     {
54         oil[i+1][j-1] = '*';
55         dfs( i+1, j-1 );
56     }
57 }
58 else if( oil[i+1][j] == '@' )
59 {
60     if( i+1 <= m )
61     {
62         oil[i+1][j] = '*';
63         dfs( i+1, j );
64     }
65 }
66 else if( oil[i+1][j+1] == '@' )
67 {
68     if( i+1 <= m && j+1 <= n )
69     {
70         oil[i+1][j+1] = '*';
71         dfs( i+1, j+1 );
72     }
73 }
74 }
75
76 int main(void)
77 {
78     while( cin >> m >> n )
79     {
80         int ans = 0;
81         if( ( m == 0 ) && ( n == 0 ) )
82         {
83             break;
84         }
85         else
86         {
87             for( int i = 0 ; i < m ; i++ )
88             {
89                 for( int j = 0 ; j < n ; j++ )
90                 {
91                     cin >> oil[i][j];
92                 }
93             }
94             for( int i = 0 ; i < m ; i++ )
95             {
96                 for( int j = 0 ; j < n ; j++ )
97                 {
98                     if( oil[i][j] == '@' )
99                     {
100                         dfs( i, j );
101                         ans++;
102                     }
103                 }
104             }
105             cout << ans << endl;
106         }
107     }
108     return 0;
109 }

```

1.2 11935

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     int num, flag = 1;
8     cin >> num;
9     while( num > 0 )
10    {

```

```

11     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) !=
20                numstr.charAt(right)) {
21                return true;
22            }
23            left++;
24            right--;
25        }
26        return false;
27    }
28    public static int reverse(int num){
29        int tot = 0, buf = 1;
30        boolean jud = true;
31        while (num > 0) {
32            tot = tot * buf + (num % 10);
33            num /= 10;
34            if(jud) buf *= 10;
35            jud = false;
36        }
37        return tot;
38    }
39    public static void main(String[] args) {
40        int flag = 0, num = 120;
41        for (int i = 2; i < 100000; i++) {
42            if (palindrome(i) && flag < num) {
43                if (prime(i) && prime(reverse(i))) {
44                    System.out.print(i);
45                    flag++;
46                    if (flag % 10 == 0) {
47                        System.out.println();
48                    }
49                    if (flag % 10 != 0 && flag !=
50                        num) {
51                        System.out.print(" ");
52                    }
53                }
54            }
55        }
56    }
57 }

```

```

51         }
52     }
53     if (flag == num && flag % 10 !=
54         0) {
55         System.out.println();
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        System.out.println();
28    }
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             {
15                 tot1 = tot1 + str.charAt(i) - '0';
16             }
17             for (int j = 1; j < str.length(); j += 2)
18             {
19                 tot2 = tot2 + str.charAt(j) - '0';
20             }
21             if( tot1 > tot2 ){
22                 judgment( tot1, tot2 );
23             }
24             else{
25                 judgment( tot2, tot1 );
26             }
27         }
28     }
29 }
30 }

```

```

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
32             of 11." );
33     }
34     else{
35         System.out.println( str + " is not a
36             multiple of 11." );
37     }
38 }
39 }
40 }

```

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
17            StringTokenizer(str, "+-*/%,()", true);
18        while(token.hasMoreTokens()){
19            String str1 = token.nextToken();
20            if( Character.isDigit(str1.charAt(0))){
21                num[flag] = str1;
22                if( flag > 0 ){
23                    System.out.print(" ");
24                }
25                System.out.print(num[flag]);
26                flag++;
27            }
28            else{
29                sign[flagg] = str1;
30                flagg++;
31            }
32        }
33        System.out.println();
34
35        for(int i = 0 ; i < sign.length ; i++ ){
36            if(sign[i] == null){
37                break;
38            }
39            else if(i > 0){
40                System.out.print(" ");
41            }
42            System.out.print(sign[i]);
43        }
44        System.out.println();
45
46        for( int i = 0 ; i < num.length ; i++ ){
47            if( num[i] == null ){
48                break;
49            }
50            BigDecimal cal = new BigDecimal(num[i]);
51            ans = ans.add(cal);
52        }
53        System.out.printf("%.3f",ans);
54        System.out.println();
55    }
56 }

```

3.6 primenumber

```

1 package com.company;
2 import java.lang.Math;
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         int num = scanner.nextInt();
9         int[] arr = new int[1000];
10        int flag = 0;
11        for (int j = 2; j < num ; 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                arr[flag] = j;
25                flag++;
26            }
27        }
28        for(int i = 0 ; i < flag ; i++){
29            int temp = i+1;
30            System.out.print(arr[i]);
31            if( temp % 10 != 0 && i != flag -1){
32                System.out.print(" ");
33            }
34
35            if( i == flag -1 && temp % 10 != 0){
36                System.out.println();
37            }
38
39            if( temp % 10 == 0){
40                System.out.println();
41            }
42        }
43    }
44 }

```

3.7 palindromeprime

```

1 package com.company;
2 import java.util.Scanner;
3 import java.lang.Math;
4 public class Main {
5
6     private static boolean prime ( int number){
7         for (int i = 2; i <= Math.sqrt(number); i++)
8         {
9             if (number % i == 0)
10            {
11                return false;
12            }
13        }
14        return true;
15    }
16    private static boolean palindrome ( int number){
17        String numstr = number + "";
18        int left = 0;
19        int right = numstr.length() - 1;
20        while (left < right) {
21            if (numstr.charAt(left) !=
22                numstr.charAt(right)) {
23                return false;
24            }
25            left++;
26            right--;

```

```

26        }
27        return true;
28    }
29    public static void main(String[] args) {
30        Scanner scanner = new Scanner(System.in);
31        while (scanner.hasNext()) {
32            int num = scanner.nextInt();
33            int flag = 0;
34            for (int i = 2; i < 100000; i++) {
35                if(num == 0){
36                    System.out.println();
37                    break;
38                }
39
40                if (palindrome(i) && flag < num) {
41                    if (prime(i)) {
42                        System.out.print(i);
43                        flag++;
44                        if (flag % 10 == 0) {
45                            System.out.println();
46                        }
47
48                        if (flag % 10 != 0 && flag !=
49                            num) {
50                            System.out.print(" ");
51                        }
52
53                        if (flag == num && flag % 10
54                            != 0) {
55                            System.out.println();
56                        }
57                    }
58                }
59            }
60        }

```

3.8 magic square

```

1 package com.company;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         while(scanner.hasNext()){
8             int n = scanner.nextInt();
9             if (n % 2 == 0){
10                System.out.println("It is not an odd
11                    number.");
12                if(scanner.hasNext()){
13                    System.out.println();
14                }
15                continue;
16            }
17            int sum = (n * ((n * n) + 1))/2;
18            System.out.println(sum);
19
20            long[][] square = new long[n][n];
21            int row = n-1;
22            int col = n/2;
23            square[row][col] = 1;
24            for (long i = 2 ; i <= n*n ; i++) {
25                if (square[(row + 1) % n][(col + 1) %
26                    n] == 0) {
27                    row = (row + 1) % n;
28                    col = (col + 1) % n;
29                }
30                else {
31                    row = (row - 1 + n) % n;
32                }
33                square[row][col] = i;
34            }
35            for (int i = 0 ; i < n ; i++) {

```

```

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

```

```

60         System.out.printf("%d\n",
61             (int)Math.sqrt(maxn));
62         boolean[] A = PrimeArray(maxn);
63         for(int i = 0 ; i < flag ; i++){
64             if(A[max[i]]){
65                 System.out.println(max[i]);
66             }
67             else{
68                 System.out.println(PrimeFactorization(max[i]));
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 =
42                        str.concat(Long.toString(flag));
43                }
44            }
45        }
46        return str;
47    }
48    public static void main(String[] args) {
49        Scanner scanner = new Scanner(System.in);
50        int maxn = 0, flag = 0;
51        int[] max = new int[1000];
52        while(scanner.hasNextInt()){
53            int N = scanner.nextInt();
54            max[flag] = N;
55            if(max[flag] > maxn){
56                maxn = max[flag];
57            }
58            flag++;
59        }
60        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 =
8             {"January", "February", "March", "April", "May", "June ",
9             "July", "August", "September", "October", "November", "December"};
10        int year = scanner.nextInt();
11        int d = scanner.nextInt();
12        int t = 0;
13        boolean leap = (year % 4 == 0 && year % 100
14            != 0) || year % 400 == 0;
15        for(int i = 0 ; i < 12 ; i++) {
16            System.out.println("      " +
17                month[i] + " " + year);
18            System.out.print("-----\n
19                Sun Mon Tue Wed Thu Fri Sat\n");
20            if( d == 7){
21                d = 0;
22            }
23            t = 0;
24            for(int j = 0 ; j < d ; j++){
25                System.out.print("    ");
26                t++;
27            }
28            int day, nd = 1;
29            switch(i+1){case 4: case 6: case 9: case
30                11: day = 30;break;
31                case 2: if(leap){day = 29;}else{day =
32                    28;}break;
33                default: day = 31;break;}
34            while(nd <= day){
35                if(t % 7 == 0 && t != 0) {
36                    System.out.println();
37                    t = 0;
38                }
39                System.out.printf("%4d", nd);
40                nd++;
41                t++;
42            }
43            System.out.println();
44            d = t;
45            if(i != 11)
46                System.out.println();
47        }
48    }
49 }

```

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","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"};
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 r1 = 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                         r1++;
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                 }
66                 for (int s = 0; s < 26; s++) {
67                     if (column[s] > 1) {
68

```

```

69         }
70     }
71 }
72 int lc = 0;
73 for (int i = 0; i < 26; i++) {
74     if (latin[i] > 0) {
75         lc++;
76     }
77 }
78
79 if (white != 0) {
80     System.out.println("");
81 }
82
83 if (r1 == 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 consecutiveFour

```

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(isConsecutiveFour(array));
18    }
19    public static boolean isConsecutiveFour(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] ==
46         values[i+3][j+3])){
47         for(int k = 1 ; k < 4 ; k++){
48             if(values[i+k][j+k] ==
49                 values[i][j])
50                 count++;
51         }
52         if(count == 3)
53             return true;
54         else
55             count = 0;
56     }
57     if((i+3 < n) && (j-3 >= 0) &&
        (values[i][j] ==
58         values[i+3][j-3])){
59         for(int k = 1 ; k < 4 ; k++){
60             if(values[i+k][j-k] ==
61                 values[i][j])
62                 count++;
63         }
64         if(count == 3)
65             return true;
66         else
67             count = 0;
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),
13                D(num1));
14            Rational n2 = new Rational(M(num2),
15                D(num2));
16            switch (co) {
17                case '+': {
18                    System.out.println(n1.outRationl()
19                        + " + " + n2.outRationl() + "
20                        = " +
21                        (n1.add(n2)).outRationl());
22                    break;
23                }
24                case '-': {
25                    System.out.println(n1.outRationl()
26                        + " - " + n2.outRationl() + "
27                        = " +
28                        (n1.sub(n2)).outRationl());
29                    break;
30                }
31                case '*': {
32                    System.out.println(n1.outRationl()
33                        + " * " + n2.outRationl() + "
34                        = " +
35                        (n1.multiply(n2)).outRationl());
36                    break;
37                }
38                case '/': {
39                    System.out.println(n1.outRationl()
40                        + " / " + n2.outRationl() + "
41                        = " +
42                        (n1.divide(n2)).outRationl());
43                    break;
44                }
45                default : {
46                }
47            }
48        }
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 /
96                this.denominator) + p.molecular * (ltemp
97                / p.denominator);
98            gtemp = gcd(mtemp, ltemp);
99            r.molecular = mtemp / gtemp;
100           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 }

```


3.16 Point

```
1 import java.util.*;
2
3 public class Main {
4     public static class Point implements
5         Comparable<Point>{
6         long x, y;
7         public Point(long x, long y){
8             this.x = x;
9             this.y = y;
10        }
11
12        @Override
13        public int compareTo(Point o) {
14            if((this.x * this.x)+(this.y * this.y) <
15                (o.x * o.x)+(o.y * o.y))
16                return -1;
17            else if((this.x * this.x)+(this.y *
18                this.y) == (o.x * o.x)+(o.y * o.y)){
19                if(this.x < o.x)
20                    return -1;
21                else if(this.x == o.x){
22                    if(this.y < o.y)
23                        return -1;
24                    else if(this.y == o.y)
25                        return 0;
26                    else
27                        return 1;
28                }
29            }
30            return 1;
31        }
32    }
33
34    public static void main(String[] args) {
35        Scanner scanner = new Scanner(System.in);
36        boolean jud = false;
37        while(scanner.hasNextLong()) {
38            if(jud)
39                System.out.println();
40            jud = true;
41            int n = scanner.nextInt();
42            Point[] p = new Point[n];
43            for(int i = 0 ; i < n ; i++){
44                p[i] = new Point(scanner.nextLong(),
45                    scanner.nextLong());
46            }
47            Arrays.sort(p);
48            long max_num = p[0].x + p[0].y;
49            int max = 0;
50            System.out.println("(" + p[0].x + "," +
51                p[0].y + ")");
52            for(int i = 1 ; i < n ; i++){
53                System.out.println("(" + p[i].x + "," +
54                    p[i].y + ")");
55                if(p[i].x + p[i].y >= max_num){
56                    max = i;
57                    max_num = p[i].x + p[i].y;
58                }
59            }
60            System.out.println("max num: " + max_num);
61            System.out.println("point:(" + p[max].x +
62                "," + p[max].y + ")");
63        }
64    }
65 }
```