

Contents

1	Graph	1
1.1	C129	1
1.2	11935	1
2	Numbers	2
2.1	CongruenceEquation	2
3	PD practice	2
3.1	practice1	2
3.2	practice2	2
3.3	practice3	2
3.4	practice4	2
3.5	HW1	3
3.6	primenumber	3
3.7	palindromeprime	4
3.8	magicsquare	4
3.9	primefactorization	4

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         }
95         for( int i = 0 ; i < m ; i++ )
96         {
97             for( int j = 0 ; j < n ; j++ )
98             {
99                 if( oil[i][j] == '@' )
100                 {
101                     dfs( i, j );
102                     ans++;
103                 }
104             }
105         }
106         cout << ans << endl;
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 PD practice

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 }

```

3.2 practice2

```

1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         for( int i = 1 ; i < 10 ; i++ ) {
7             for( int j = 1 ; j < 10 ; j++ ) {
8                 System.out.print( i+ " * "+ j+ " =");
9                 if( i * j < 10 ){
10                     System.out.print(" ");
11                 }
12                 else{
13                     System.out.print(" ");
14                 }
15                 int ans = i * j;
16                 if( j == 9 )
17                 {
18                     System.out.print(ans);
19                 }
20                 else
21                 {
22                     System.out.print( ans+ " " );
23                 }
24             }
25             System.out.println();
26         }
27     }
28 }
29 }

```

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

```

```

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

```

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--;
27        }
28        return true;
29    }
30    public static void main(String[] args) {
31        Scanner scanner = new Scanner(System.in);
32        while (scanner.hasNext()) {
33            int num = scanner.nextInt();
34            int flag = 0;
35            for (int i = 2; i < 100000; i++) {
36                if(num == 0){
37                    System.out.println();
38                    break;
39                }
40
41                if (palindrome(i) && flag < num) {
42                    if (prime(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 !=
50                            num) {
51                            System.out.print(" ");
52                        }
53
54                        if (flag == num && flag % 10
55                            != 0) {
56                            System.out.println();
57                        }
58                    }
59                }
60            }
61        }
62    }
63 }

```

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
36             for (int i = 0 ; i < n ; i++) {
37                 for (int j = 0 ; j < n ; j++) {
38                     System.out.printf("%5d",
39                         square[i][j]);
40                 }
41                 System.out.println();
42             }
43             if(scanner.hasNext()){
44                 System.out.println();
45             }
46         }
47     }
48 }

```

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 }

```

```
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 + " ");
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 }
```