Contents

1 Graph

1.1 C129

```cpp
#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

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

using namespace std;

int main()
{
    int num, flag = 1;
    cin >> num;
    while( num > 0 )
    {
        int n, ans = 0;
        char map[100][100] = {0};
        cin >> n;
        for( int i = 0 ; i < n ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
```

```
            {
                cin >> map[i][j];
            }
        }
        for( int i = 0 ; i < n ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
            {
                if( map[i][j] == 'x' )
                {
                    ans++;
                }
            }
        }
        cout << "Case " << flag << ": " << ans <<endl;
        num--;
        flag++;
    }
    return 0;
}
```

2 Numbers

2.1 CongruenceEquation

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

using namespace std;

long long Mode(long long a, long long n, long long m)
{
    long long sum = 1;
    for( ; n ; n >>= 1 )
    {
        if( n & 1 )
        {
            sum = ( sum * a ) % m;
        }
        a = ( a * a ) % m;
    }
    return sum;
}

int main(void)
{
    int a, b, p, x, ans = 0;
    cin >> a >> b >> p >> x;
    for( int i = 1 ; i < x + 1 ; i++ )
    {
        int n;
        n = i % p;
        n = n * Mode( a, i, p);
        if( n % p == b % p )
        {
            ans++;
        }
    }
    cout << ans <<endl;
    return 0;
}
```

3 PD practice

3.1 practice1

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

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int m = n-1;
        for( int i = 1 ; i <= 2*n-1 ; i=i+2 ) {
            for( int j = m ; j > 0 ; j-- ) {
                System.out.print(" ");
            }
            m--;
            for (int t = 0; t < i; t++) {
                System.out.print("*");
            }
            System.out.println();
        }

    }
}
```

3.2 practice2

```
package com.company;

public class Main {

    public static void main(String[] args) {
        for( int i = 1 ; i < 10 ; i++ ) {
            for( int j = 1 ; j < 10 ; j++ ) {
                System.out.print( i+ " * "+ j+ " =");
                if( i * j < 10 ){
                    System.out.print("  ");
                }
                else{
                    System.out.print(" ");
                }
                int ans = i * j;
                if( j == 9 )
                {
                    System.out.print(ans);
                }
                else
                {
                    System.out.print( ans+ " " );
                }
            }
            System.out.println();
        }

    }
}
```

3.3 practice3

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

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        if( n >= 2 ){
            System.out.print(2);
        }
        for (int j = 3; j < n ; j++)
        {
            boolean answer = true;
            for (int i = 2; i <= Math.sqrt(j) ; i++)
            {
                if (j % i == 0)
                {
                    answer = false;
                    break;
                }
            }
            if (answer)
            {
```

```
                System.out.print( " "+ j );

            }
        }
        System.out.println();
    }
}
```

3.4 practice4

```
package com.company;
import java.util.Scanner;
public class Main {
    private static String str;
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while( scanner.hasNext() ) {
            str = scanner.next();
            if(str.equals("0")){
                break;
            }
            int tot1 = 0, tot2 = 0;
            for (int i = 0; i < str.length(); i += 2)
                {
                tot1 = tot1 + str.charAt(i) - '0';
            }
            for (int j = 1; j < str.length(); j += 2)
                {
                tot2 = tot2 + str.charAt(j) - '0';
            }
            if( tot1 > tot2 ){
                judgment( tot1, tot2 );
            }
            else{
                judgment( tot2, tot1 );
            }
        }
    }

    public static void judgment( int a, int b ){
        int judge = a - b;
        if( judge % 11 == 0 ){
            System.out.println( str +" is a multiple
                of 11." );
        }
        else{
            System.out.println( str +" is not a
                multiple of 11." );
        }
    }
}
```

3.5 HW1

```
package com.company;
import java.math.BigDecimal;
import java.util.Scanner;
import java.util.StringTokenizer;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String str = scanner.next();
        BigDecimal ans = new BigDecimal(0);
        String[] num = new String[50];
        String[] sign = new String[50];

        int flag = 0, flagg = 0;

        StringTokenizer token = new
            StringTokenizer(str,"+-*/%,()",true);
        while(token.hasMoreTokens()){
            String str1 = token.nextToken();
            if( Character.isDigit(str1.charAt(0))){
                num[flag] = str1;
                if( flag > 0 ){
                    System.out.print(" ");
                }
                System.out.print(num[flag]);
                flag++;
            }
            else{
                sign[flagg] = str1;
                flagg++;
            }
        }
        System.out.println();

        for(int i = 0 ; i < sign.length ; i++ ){
            if(sign[i] == null){
                break;
            }
            else if(i > 0){
                System.out.print(" ");
            }
            System.out.print(sign[i]);
        }
        System.out.println();

        for( int i = 0 ; i < num.length ; i++ ){
            if( num[i] == null ){
                break;
            }
            BigDecimal cal = new BigDecimal(num[i]);
            ans = ans.add(cal);
        }
        System.out.printf("%.3f",ans);
        System.out.println();
    }
}
```

3.6 primenumber

```
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

```
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--;
        }
        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

```
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++) {
                for (int j = 0 ; j < n ; j++) {
                    System.out.printf("%5d",
                        square[i][j]);
                }
                System.out.println();
            }
            if(scanner.hasNext()){
                System.out.println();
            }
        }
    }
}
```

3.9 primefactorization

```
import java.util.Scanner;

public class Main {
    private static boolean[] PrimeArray(long N){
        boolean[] A = new boolean[(int)N+1];
        A[0] = true;
        A[1] = true;
        for(long j = 2; j <= N ; j++) {
            boolean judge = true;
            for (int i = 2; i <= Math.sqrt(j); i++) {
                if (j % i == 0) {
                    judge = false;
                    A[(int) j] = false;
                    break;
                }
            }
            if(judge){
                A[(int)j] = true;
            }
        }
        return A;
    }
```

```
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  }
```