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 JAVApractice

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 emirp

```
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 true;
            }
            left++;
            right--;
        }
        return false;
    }
    public static int reverse(int num){
        int tot = 0, buf = 1;
        boolean jud = true;
        while (num > 0) {
            tot = tot * buf + (num % 10);
            num /= 10;
            if(jud) buf *= 10;
            jud = false;
        }
        return tot;
    }
    public static void main(String[] args) {
        int flag = 0, num = 120;
        for (int i = 2; i < 100000; i++) {
            if (palindrome(i) && flag < num) {
                if (prime(i) && prime(reverse(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();
                    }
                }
            }
        }
    }
}
```

3.3 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

```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--;
        }
        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++) {
                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;
    }
    private static String PrimeFactorization(long N){
        String str = "";
        boolean jud = false;
        for(long i = 2 ; N > 1 ; i++) {
            int flag = 1;
            if(N % i == 0) {
                if(jud) {
                    str = str + " * ";
                }
                N = N / i;
                str = str.concat(Long.toString(i));
                jud = true;
                while (N % i == 0) {
                    N = N / i;
                    flag++;
                }
                if(flag > 1){
                    str = str + "^";
                    str =
                        str.concat(Long.toString(flag));
                }
            }
        }
        return str;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int maxn = 0, flag = 0;
        int[] max = new int[1000];
        while(scanner.hasNextInt()){
            int N = scanner.nextInt();
            max[flag] = N;
            if(max[flag] > maxn){
                maxn = max[flag];
            }
            flag++;
        }
        System.out.print(maxn + " ");
        System.out.printf("%d\n",
            (int)Math.sqrt(maxn));
        boolean[] A = PrimeArray(maxn);
        for(int i = 0 ; i < flag ; i++){
            if(A[max[i]]){
                System.out.println(max[i]);
            }
            else{
                System.out.println(PrimeFactorization(max[i]));
            }
        }
    }
}
```

3.10 calendar

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String[] month =
            {"January","February","March","April","May","June",
        int year = scanner.nextInt();
        int d = scanner.nextInt();
        int t = 0;
        boolean leap = (year % 4 == 0 && year % 100
            != 0) || year % 400 == 0;
        for(int i = 0 ; i < 12 ; i++) {
            System.out.println("          " +
                month[i] + " " + year);
            System.out.print("-----------------------------\n
                Sun Mon Tue Wed Thu Fri Sat\n");
            if( d == 7){
                d = 0;
            }
            t = 0;
            for(int j = 0 ; j < d ; j++){
                System.out.print("    ");
                t++;
            }
            int day, nd = 1;
            switch(i+1){case 4: case 6: case 9: case
                11: day = 30;break;
                case 2: if(leap){day = 29;}else{day =
                    28;}break;
                default: day = 31;break;}
            while(nd <= day){
                if(t % 7 == 0 && t != 0) {
                    System.out.println();
                    t = 0;
                }
                System.out.printf("%4d", nd);
                nd++;
                t++;
            }
            System.out.println();
            d = t;
            if(i != 11)
                System.out.println();
        }
    }
}
```

3.11 latinsquare

```
import java.util.Scanner;
public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int num, white = 0;
        if(( num= scanner.nextInt()) > 0 )
        {
            int[] latin = new int[26] ;
            String[][] square = new String[num][num];
```

```
            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"};
            int flag = 0;
            boolean ha = true;
            for(int i = 0 ; i < num ; i++)
            {
                int flagg = 0;
                while( scanner.hasNext() && flagg <
                    num)
                {
                    square[i][flagg] = scanner.next();
                    if((square[i][flagg].charAt(0) -
                        '0') >
                        (eng[num].charAt(0)-'0')){
                        System.out.printf("Wrong
                            input: the letters must
                            be from %s to %s\n",
                            eng[0], eng[num-1]);
                        ha = false;
                        break;
                    }
                    flagg++;
                }
                if(flagg != num && ha){
                    System.out.printf("Wrong input:
                        you need to enter exactly %d
                        letters\n", num);
                    ha = false;
                    break;
                }
                flag++;
            }
            if(flag != num && ha){
                System.out.printf("Wrong input: you
                    need to enter exactly %d
                    letters\n", num);
                ha = false;
            }
            if(ha) {
                int rl = 0;
                for (int i = 0; i < num; i++) {
                    int[] row = new int[26];
                    for (int j = 0; j < num; j++) {
                        for (int t = 0; t < 26; t++) {
                            if
                                (square[i][j].equals(eng[t]))
                                {
                                row[t]++;
                                latin[t]++;
                            }
                        }
                    }
                    for (int s = 0; s < 26; s++) {
                        if (row[s] > 1) {
                            rl++;
                        }
                    }
                }
                int cl = 0;
                for (int i = 0; i < num; i++) {
                    int[] column = new int[26];
                    for (int j = 0; j < num; j++) {
                        for (int t = 0; t < 26; t++) {
                            if
                                (square[j][i].equals(eng[t]))
                                {
                                column[t]++;
                            }
                        }
                        for (int s = 0; s < 26; s++) {
                            if (column[s] > 1) {
                                cl++;
                            }
                        }
                    }
                }
                int lc = 0;
                for (int i = 0; i < 26; i++) {
                    if (latin[i] > 0) {
                        lc++;
                    }
                }

                if (white != 0) {
                    System.out.println("");
                }

                if (rl == 0 && cl == 0 && lc == num) {
                    System.out.println("The input
                        array is a Latin square");
                } else {
                    System.out.println("The input
                        array is not a Latin square");
                }
                white++;
            }
        }
    }
}
```