Fibonacci

```
import java.util.Scanner;
class lastdigit {
    void fibonacci(int f[]) {
        f[0] = 0;
        f[1] = 1;

        for(int i = 2; i <= 59; i++) {
            f[i] = (f[i - 1] + f[i - 2]) % 10;
        }
    }

    int lastDigit(int n) {
        int f[] = new int[60];
        fibonacci(f);
        int ans = n % 60;
        return f[ans];
    }

    public static void main(String[] arg) {
        lastdigit ob = new lastdigit();
        Scanner input = new Scanner(System.in);
        int num = input.nextInt();
        System.out.println(ob.lastDigit(num));
    }
}
```

Number System

```
import java.util.Scanner;
import java.lang.Math;
class numsystem {
    static void printSequence (int a, int b) {   ///  1234  = 1*10^3 + 2*10^2 + 3*10^1 + 4*10^0 =
1234
        int ans = 0;
        int i = 0;
        while(a>0) {
            int count = a%10;
            a = a/10;
            ans = ans + count * (int)Math.pow(b,i);
            i++;
        }
        System.out.println(ans);
    }

    public static void main(String[] arg) {
        Scanner input = new Scanner(System.in);
        int basevalue = input.nextInt();
        int num = input.nextInt();
        printSequence(num, basevalue);
    }
}
```

Collatz Sequence

```
import java.util.Scanner;
class collatz {
    static void printSequence (int n) {
        while(n != 1) {
            System.out.print(n + " ");
            if(n % 2 != 0) {
                n = 3 * n +1;
            }
            else {
                n = n / 2;
            }
        }
        System.out.print(n);
    }

    public static void main(String[] arg) {
        Scanner input = new Scanner(System.in);
        int num = input.nextInt();
        printSequence(num);
    }
}
```

Date difference

```
public class DateDiff
{
    static int countLeapYears(int d, int m, int y) {
        int count = y;
        if(m <= 2) {
            count--;
        }
        return count/4 - count/100 + count/400;
    }

     public static void main(String[] args) {
     int d1, d2, m1, m2, y1, y2;
     d1 = Integer.parseInt(args[0]);
     m1 = Integer.parseInt(args[1]);
     y1 = Integer.parseInt(args[2]);
     d2 = Integer.parseInt(args[3]);
     m2 = Integer.parseInt(args[4]);
     y2 = Integer.parseInt(args[5]);

     int months[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    int ans1 = y1 * 365 + d1;
    for(int i = 0; i < m1 - 1; i++ ) {
        ans1 += months[i];
    }
    ans1 += countLeapYears(d1, m1, y1);

    int ans2 = y2 * 365 + d2;
    for(int i = 0; i < m2 - 1; i++ ) {
        ans2 += months[i];
    }
    ans2 += countLeapYears(d2, m2, y2);

     System.out.println(ans2-ans1);

     }
}
```

Check Anagram

```
import java.util.Arrays;
import java.util.Scanner;
class checkanagram {

    static void anagram(char[] arr1, char[] arr2) {
        if(arr1.length != arr2.length) {
          System.out.println("-1");
          return;
        }
        int count = 0;
        for(int i = 0; i < arr1.length; i++) {
          if(arr1[i] == arr2[i]) {
            count++;
          }
        }
        boolean ans = true;
        Arrays.sort(arr1);
        Arrays.sort(arr2);

        for(int i = 0; i < arr1.length; i++) {
          if(arr1[i] != arr2[i]) {
            ans = false;
          }
        }

        if(ans) {
          System.out.println(count);
        }
        else {
          System.out.println("-1");
        }
      }
   public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String str1 = input.nextLine();
    String str2 = input.nextLine();
    char[] arr1 = str1.toCharArray();
    char[] arr2 = str2.toCharArray();
    anagram(arr1, arr2);
   }
}
```

Biggest palindrome

```
import java.util.Scanner;
public class biggestPalindrome
{
   static void biggestPalString(String str) {
      if(str.length() < 2) {
         System.out.println(str);
         System.out.println(str.length());
         return;
      }
      int l, h, count = 1, place = 0;;
      for (int i = 0; i < str.length(); i++) {
         l = i - 1;
         h = i + 1;
         while(l >= 0 && str.charAt(l) == str.charAt(i))
            l--;

         while(h < str.length() && str.charAt(h) == str.charAt(i))
            h++;

         while(l >= 0 && h < str.length() && str.charAt(l) == str.charAt(h)) {
              l--;
              h++;
         }
         int lenght = h - l - 1;
         if(count < lenght) {
            count = lenght;
            place = l + 1;
         }
      }
         System.out.println(str.substring(place, place + count));
         System.out.println(count);

   }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String str = input.nextLine();
        biggestPalString(str);
    }
}
```

Spiral Matrix

```
import java.util.*;
```

```
class spiralMatrix
{
    static void printSpiralMatrix(int n)
    {
       int[][] a = new int[n][n];
        int top = 0, bottom = n - 1, l = 0, r = n - 1, count = 1;

        while(true)
        {
            if(l > r)
                break;

            for(int i = l; i <= r; i++)
                a[top][i] = count++;
            top++;

            if(top > bottom)
                break;

            for(int i = top; i <= bottom; i++)
                a[i][r] = count++;
            r--;

            if(l > r)
                break;

            for(int i = r; i >= l; i--)
                a[bottom][i] = count++;
            bottom--;

            if(top > bottom)
                break;

            for(int i = bottom; i >= top; i--)
                a[i][l] = count++;
            l++;
        }

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
                System.out.print(a[i][j] + "\t");
            System.out.println();
        }
    }

    public static void main (String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        printSpiralMatrix(n);
    }
}
```

Magic Square

```
import java.util.*;
public class magicSquare
{
   static void magicSquarePrint(int n) {
   int a[][] = new int[n][n];
   int r = 0, c = n/2;

   for(int i = 1; i <= n*n; i++ ) {

      a[r][c] = i;

      if(i % n == 0) r++;

      else{

         if(r == 0) r = n - 1;

         else r--;

         if(c == (n - 1)) c = 0;

         else c++;

      }

   }
   for(int i = 0; i < n; i++) {
         for(int j = n - 1; j >=0 ; j--)
            System.out.print(a[i][j] + "  ");
            System.out.println();
        }
    }

     public static void main(String[] args) {
         Scanner input = new Scanner(System.in);
         int n = input.nextInt();
         magicSquarePrint(n);
     }
}
```

Largest Repeated Substring

```
import java.util.*;
public class largestrepeated
{
    static void largestrepeatedsubstring(String str) {
        int n = str.length();
        int dp[][] = new int[n + 1][n + 1];
        int count = 0, in = 0;
        for(int i = 1; i <= n; i++) {
            for(int j = i + 1; j <= n; j++) {
                if(str.charAt(i - 1) == str.charAt(j - 1) && j - i > dp[i - 1][j - 1]) {
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                    if(count < dp[i][j]) {
                        count = dp[i][j];
                        in = Math.max(i, in);
                    }
                }
                else {
                    dp[i][j] = 0;
                }
            }
        }
        System.out.println(str.substring(in - count, in));
    }
     public static void main(String[] args) {
         Scanner input = new Scanner(System.in);
         String str = input.nextLine();
         largestrepeatedsubstring(str);
     }
}
```

Find word

```
import java.util.*;

public class findWord {

   public static int solve(int[][] vis, char[][] v, String words[], int i, int j, int idx, int k) {

      if (idx >= words[k].length())
         return 1;

      if (i >= v.length || j >= v[0].length || i < 0 || j < 0 || v[i][j] != words[k].charAt(idx) || vis[i][j] == 1)
         return 0;

      vis[i][j] = 1;
      int a = 0;

      a |= solve(vis, v, words, i+1, j, idx + 1, k);
      a |= solve(vis, v, words, i+1, j-1, idx + 1, k);
      a |= solve(vis, v, words, i+1, j+1, idx + 1, k);
      a |= solve(vis, v, words, i, j+1, idx + 1, k);
      a |= solve(vis, v, words, i-1, j+1, idx + 1, k);
      a |= solve(vis, v, words, i-1, j, idx + 1, k);
      a |= solve(vis, v, words, i-1, j-1, idx + 1, k);
      a |= solve(vis, v, words, i, j-1, idx + 1, k);

      vis[i][j] = 0;

      return a;
   }

   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);

      int n = sc.nextInt();
      char[][] v = new char[n][n];
      int[][] vis = new int[n][n];

      for (int i = 0; i < n; i++) {
         for (int j = 0; j < n; j++) {
            v[i][j] = sc.next().charAt(0);
```

```
                vis[i][j] = 0;
            }
        }

        String words[] = { "APPLE", "BANANA", "CHERRY", "GRAPES", "LEMON", "ORANGE",
"TOMATO" };

        for (int k = 0; k < words.length; k++) {
            int a=0;
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    if (v[i][j] == words[k].charAt(0)) {
                        a=solve(vis, v, words, i, j, 0, k);
                        if(a==1){
                            System.out.print(i+" "+j+" ");
                            break;
                        }
                    }
                }
                if(a==1)break;
            }
        }

    }
}
```