```
public class LoanCalc {

    static double epsilon = 0.001;
    static int iterationCounter;

    public static void main(String[] args) {

        double loan = Double.parseDouble(args[0]);
        double rate = Double.parseDouble(args[1]);
        int n = Integer.parseInt(args[2]);
        System.out.println("Loan sum = " + loan + ", interest rate = " + rate + "%, periods = " + n);

        System.out.print("Periodical payment, using brute force: ");
        System.out.printf("%.2f", bruteForceSolver(loan, rate, n, epsilon));
        System.out.println();
        System.out.println("number of iterations: " + iterationCounter);

        System.out.print("Periodical payment, using bi-section search: ");
        System.out.printf("%.2f", bisectionSolver(loan, rate, n, epsilon));
        System.out.println();
        System.out.println("number of iterations: " + iterationCounter);
    }


    public static double bruteForceSolver(double loan, double rate, int n, double epsilon) {

            iterationCounter = 0;
            double g = loan / n;
            double increment = 0.001;

            while((g < loan) && (endBalance(loan, rate, n, g)) >= epsilon) {

                g += increment;
                iterationCounter++;

            }


        return g;
    }

    public static double bisectionSolver(double loan, double rate, int n, double epsilon) {

            iterationCounter = 0;
            double L = loan / n;
            double H = loan;
            double g = (L + H)/2.0;

            while(H-L >= epsilon){

                if(endBalance(loan, rate, n, g) * endBalance(loan, rate, n, L) > 0) {

                    L = g;

                } else {

                    H = g;
                }

                iterationCounter++;
                g = (L + H) / 2;
            }


        return g;
    }

    private static double endBalance(double loan, double rate, int n, double payment) {


        for(int i = 0; i < n; i++){

            loan -= payment;
            loan *= (1 + rate/100);

        }

        return loan;
    }
}
```

```
public class LowerCase {
    public static void main(String[] args) {

        String str = args[0];
        System.out.println(lowerCase(str));


    }

    public static String lowerCase(String s) {

        String newS = "";

        for (int i = 0; i < s.length(); i++){

            char c = s.charAt(i);

            if ((c >= 65) && (c <= 90)){

                c += 32;

            }

             newS += c;

        }

        return newS;
    }
}
```

```
public class UniqueChars {
    public static void main(String[] args) {

        String str = args[0];
        System.out.println(uniqueChars(str));

    }

    public static String uniqueChars(String s) {

        String newS = "";


        for (int i = 0; i < s.length(); i++){


           if ((newS.indexOf(s.charAt(i)) == -1) || (s.charAt(i) == ' ')){

                newS += s.charAt(i);

            }

        }

        return newS;
    }
}
```

```
public class Calendar {

    static int dayOfMonth = 1;
    static int month = 1;
    static int year = 1900;
    static int dayOfWeek = 2;

    public static void main(String args[]) {

        int pickedYear = Integer.parseInt(args[0]);
        advance(pickedYear);

     }

     private static void advance(int s) {

        while(year <= s){

            for (int month = 1; month <= 12; month++) {

                for (int dayOfMonth = 1; dayOfMonth <= nDaysInMonth(month, year);
                        dayOfMonth++){

                        String date = (dayOfMonth + "/" + month + "/" + year);

                        if (dayOfWeek == 8){

                            dayOfWeek = 1;
                        }

                        if (year == s && dayOfWeek == 1){

                            System.out.println(date + " Sunday");

                        } else if (year == s){

                            System.out.println(date);

                        }

                        dayOfWeek++;

                    }
                }
                year++;

            }
        }

    private static boolean isLeapYear(int year) {

        if ((year % 400 == 0) || ((year % 4 == 0) && (year % 100 != 0))) {

            return true;
        }

        return false;
    }

    private static int nDaysInMonth(int month, int year) {

        if ((isLeapYear(year)) && (month == 2)){

            return 29;

        }

        else if (month == 2){

            return 28;

        }

        else if((((double) month % 2 == 0) && (month <= 6)) || ((double) month % 2 != 0 && month
                        > 7)) {

            return 30;

        }

        else {

            return 31;

        }
    }
}
```