```java
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;
        while (endBalance(loan, rate, n, g)>0)
        {
            g = g + epsilon;
            iterationCounter++;
        }
        return g;
    }

    public static double bisectionSolver(double loan, double rate, int n, double epsilon)
    {
        iterationCounter = 0;
        double max = loan;
        double min = 0;
        double g = (max + min)/2;
        while (max - min > epsilon)
        {
            if ((endBalance(loan, rate, n, max)* endBalance(loan, rate, n, g)) < 0)
            {
                min = g;
            }
            else
            {
                max = g;
            }
            g = (max + min)/2;
            iterationCounter++;

        }
        return g;

    }

    private static double endBalance(double loan, double rate, int n, double payment)
    {
        double balance = 0;
        double current = loan;
        for (int i = 1; i <= n; i++)
        {
            balance = (current-payment)*(1 + (rate/100));
            current = balance;
        }
        return balance;
    }
}
```

```
class LowerCase
{
    public static void main(String[] args)
    {
        String str = args[0];
        System.out.println(lowerCase(str));
    }
    public static String lowerCase(String s)
    {
       String result = "";
       for (int i=0; i<s.length(); i++)
       {
        char c = s.charAt(i);
        if ((c >= 'A') && (c <= 'Z'))
        {
            result += (char)(c + 32);
        }
        else
        {
            result += c;

        }
       }
        return result;
    }
}
```

```java
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 result = "";
        for (int i = 0; i < s.length(); i++)
        {
            char char1 = s.charAt(i);
            boolean isUnique = true;
            if (s.indexOf(" ", i) == i)
            {
                result = result + " ";
            }
            for (int j = 0; j < result.length(); j++)
            {
                char char2 = result.charAt(j);
                if (char1 == char2)
                {
                   isUnique = false;
                   break;
                }
            }
            if (isUnique)
            {
               result += char1;
            }

        }
        return result;
    }
}
```

```java
public class Calendar
{
    static int dayOfMonth = 1;
    static int month = 1;
    static int year = 1900;
    static int dayOfWeek = 2;     // 1.1.1900 was a Monday
    static int nDaysInMonth = 31; // Number of days in January

    public static void main(String args[])
    {
        int currentYear = Integer.parseInt(args[0]);

        while(year< currentYear)
        {

            while (month<13)
            {
                nDaysInMonth = nDaysInMonth(month, year);
                for (int j = 1; j <= nDaysInMonth; j++)
                {
                    if (dayOfWeek == 7)
                    {
                        dayOfWeek = 1;
                    }
                    else
                    {
                        dayOfWeek ++;
                    }
                }
                month ++;
            }

            year++;
            month = 1;
        }
        month = 1;
        for (int k =1; k <= 12; k++)
        {
            advance(dayOfMonth, month, currentYear);
            month++;
        }
        if (month == 13)
        {
            month = 1;
        }
        year++;

     }


     private static void advance(int dayOfMonth, int month, int year)
    {
        nDaysInMonth = nDaysInMonth(month, year);
        for (int j = 1; j <= nDaysInMonth; j++)
        {
            System.out.print(dayOfMonth + "/" + month + "/" + year);
            if (dayOfWeek == 1)
            {
                System.out.print(" Sunday");
            }
            System.out.println();
            if (dayOfWeek == 7)
            {
                dayOfWeek = 1;
            }
            else
            {
                dayOfWeek ++;
            }
            dayOfMonth ++;
        }
        dayOfMonth = 1;
    }

    private static boolean isLeapYear(int year)
    {
        if (year % 4 == 0)
        {
            if (year % 100 == 0)
            {
                return year % 400 == 0;
            }
            else
            {
                return true;
            }
        }
        else
        {
            return false;
        }
    }

    private static int nDaysInMonth(int month, int year)
    {
        int days = 0;
        if (month == 2)
        {
            if (isLeapYear(year))
            {
                days = 29;
            }
            else days = 28;
        }
        else if ((month == 4) || (month == 6) || (month == 9) || (month == 11))
        {
            days = 30;
        }
        else days = 31;
        return days;
    }
}
```