

```

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;
        double f = endBalance(loan, rate, n, g);
        while (f >= epsilon && f >= 0){
            g += increment;
            f = endBalance(loan, rate, n, g);
            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;
        double f = endBalance(loan, rate, n, g);
        while((h - l) > epsilon){
            if (endBalance(loan, rate, n, g)*endBalance(loan, rate, n, l) > 0){
                l = g;
            } else {
                h = g;
            }
            g = (l + h) / 2;
            iterationCounter++;
        }
        return g;
    }

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

```

```
        for (int i = 1; i <= n; i++){  
            loan = (loan - payment) * (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 answer = "";  
        for(int i = 0; i < s.length(); i++){  
            char chr = s.charAt(i);  
            if ( chr >= 'A' && chr <= 'Z'){  
                char chrLow = ((char)(chr + 32));  
                chr = chrLow;  
                answer += chr;  
            }  
            else{  
                answer += chr;  
            }  
        }  
        return answer;  
    }  
}
```

```
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 chrUnique = "";  
        for( int i = 0; i < s.length(); i++){  
            char chr = s.charAt(i);  
            if (s.indexOf(chr) == i || chr == ' '){  
                chrUnique += s.charAt(i);  
            }  
        }  
        return chrUnique;  
    }  
}
```

```

public class Calendar0 {
    public static void main(String args[]) {
        int year = Integer.parseInt(args[0]);
        isLeapYearTest(year);
        nDaysInMonthTest(year);
    }

    private static void isLeapYearTest(int year) {
        String commonOrLeap = "common";
        if (isLeapYear(year)) {
            commonOrLeap = "leap";
        }
        System.out.println(year + " is a " + commonOrLeap + " year");
    }

    private static void nDaysInMonthTest(int year) {
        for(int i = 1; i <= 12; i++) {
            System.out.println("Month " + i + " has " + nDaysInMonth(i, year) +
" days");
        }
    }

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

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

```

```

public class Calendar1 {
    static int dayOfMonth = 1;
    static int month = 1;
    static int year = 1900;
    static int dayOfWeek = 2;
    static int nDaysInMonth = 31;

    public static void main(String args[]) {
        int debugDaysCounter = 0;
        int sundaysCount = 0;
        while (year < 2000) {
            System.out.print(dayOfMonth + "/" + month + "/" + year);
            if(dayOfWeek == 1) {
                System.out.print(" sunday");
                if(dayOfMonth == 1) {
                    sundaysCount++;
                }
            }
            System.out.println("");
            advance();
            debugDaysCounter++;
            if (debugDaysCounter == 36500) {
                break;
            }
        }
        System.out.print("During the 20th century, " + sundaysCount + "
Sundays fell on the first day of the month");
    }

    private static void advance() {
        dayOfWeek++;
        if(dayOfWeek > 7) {
            dayOfWeek = 1;
        }
        dayOfMonth++;
        if(dayOfMonth > nDaysInMonth) {
            month++;
            dayOfMonth = 1;
            if(month > 12) {
                year++;
                month = 1;
            }
        }
        nDaysInMonth = nDaysInMonth(month, year);
    }

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

```

```
}

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

}
```

```

public class Calendar {
    static int dayOfMonth = 1;
    static int month = 1;
    static int year = 1900;
    static int dayOfWeek = 2;
    static int nDaysInMonth = 31;

    public static void main(String args[]) {
        int givenYear = Integer.parseInt(args[0]);
        while (year < givenYear) {
            advance();
        }
        while (year == givenYear) {
            System.out.print(dayOfMonth + "/" + month + "/" + year);
            if(dayOfWeek == 1) {
                System.out.print(" Sunday");
            }
            advance();
            System.out.println("");
        }
    }

    private static void advance() {
        dayOfWeek++;
        if(dayOfWeek > 7) {
            dayOfWeek = 1;
        }
        dayOfMonth++;
        if(dayOfMonth > nDaysInMonth) {
            month++;
            dayOfMonth = 1;
            if(month > 12) {
                year++;
                month = 1;
            }
        }
        nDaysInMonth = nDaysInMonth(month, year);
    }

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

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

```



```
        return 29;
    } else {
        return 28;
    }
}
return 31;
}
```