```
public class LoanCalc {

        static double epsilon = 0.001;  // The computation tolerance (estimation error)
        static int iterationCounter;    // Monitors the efficiency of the calculation

        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 payment = loan / n;
                while(endBalance(loan, rate, n, payment) > epsilon) {
                        iterationCounter++;
                        payment = payment + epsilon;
                }
                return payment;
           }
public static double bisectionSolver(double loan, double rate, int n, double epsilon) {
        iterationCounter = 0;
        double right = loan;
        double left = 0;
        double payment = (right + left) / 2;
        while((right - left) > epsilon) {
                iterationCounter++;
                if ((endBalance(loan, rate, n, payment) * endBalance(loan, rate, n, left))
                > 0) {
                        left = payment;
                } else {
                        right = payment;
                }
                payment = (right + left) / 2;
        }
        return payment;
   }

   private static double endBalance(double loan, double rate, int n, double payment) {
        for (int i = 0; i < n; i++) {
        loan  = (loan - payment) * ((100 + 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) {
                int stringLength = s.length();
                String newSentence = "";
                for(int i = 0; i < stringLength; i++) {
                        char currentChar = s.charAt(i);
                        if(currentChar > 64 && currentChar < 91) {
                                char lowCase = (char)(currentChar + 32);
                                currentChar = lowCase;
                                newSentence = newSentence + currentChar;
                        } else {
                        newSentence = newSentence + currentChar;
                        }
                }
         return newSentence;
        }
}
```

```
public class UniqueChars {
        public static void main(String[] args) {
              String str = args[0];
              System.out.println(uniqueChars(str));
        }

        public static String uniqueChars(String s) {
                int stringLength = s.length();
                String newSentence = "";
                for(int i = 0; i < stringLength; i++) {
                        char currentChar = s.charAt(i);
                        if(s.indexOf(currentChar) == i || currentChar == ' ') {
                                newSentence = newSentence + currentChar;
                        }
                }
         return newSentence;
        }
}
```

```
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;
        }
}
```