

פתרון שיעורי בית 3

1)

```
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) {
        // Gets the loan data
        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);

        // Computes the periodical payment using brute force search
        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);

        // Computes the periodical payment using bisection search
        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) {
        double payment = loan/n;
        iterationCounter = 0;
        while(endBalance(loan, rate, n, payment) > epsilon){
            payment += epsilon;
            iterationCounter++;
        }
        return payment;
    }

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

```

    }
    else{
        h = payment;
    }
    payment = (l + h) / 2;
    iterationCounter++;
}
return payment;
}

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

```

2)

```
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 change = "";  
        for (int i = 0 ; i < s.length() ; i++ ) {  
            char check = s.charAt(i);  
            if( (int)check >= 65 && (int)check <= 90){  
                check += 32;  
                change += check;  
            }  
            else{  
                change += s.charAt(i);  
            }  
        }  
        return change;  
    }  
}
```

3)

```
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 check = "";
        int count = 0;
        for (int i = 0; i < s.length() ; i++) {
            for (int j = 0; j < check.length() ; j++) {
                if(s.charAt(i) != ' ' && s.charAt(i) != check.charAt(j)){
                    count++;
                }
            }
            if(count == check.length()){
                check += s.charAt(i);
            }
            if(s.charAt(i) == ' '){
                check += ' ';
            }
            count = 0;
        }
        return check;
    }
}
```

4)

```
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;
        }
        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;
                return 28;
            }
            return 31;
        }
    }
}
```

5)

```
public class Calendar1 {
    // Starting the calendar on 1/1/1900
    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 sunday = 0;
        while (year < 2000) {
            if(dayOfWeek == 1 && dayOfMonth == 1){
                System.out.println(dayOfMonth + "/" + month + "/" + year + "
Sunday");
                sunday++;
            }
            else{
                System.out.println(dayOfMonth + "/" + month + "/" + year);
            }
            advance();
        }
        System.out.println("During the 20th century, " + sunday + " Sundays fell
on the first day of the month");
    }

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


6)

```
public class Calendar{
    // Starting the calendar on 1/1/1900
    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 givenyear = Integer.parseInt(args[0]);
        while (year < 2025) {
            if(year == givenyear){
                while(year == givenyear)
                {
                    if(dayOfWeek == 1){
                        System.out.println(dayOfMonth + "/" + month + "/" + year + "
Sunday");
                    }
                    else{
                        System.out.println(dayOfMonth + "/" + month + "/" + year);
                    }
                    advance();
                }
            }
            advance();
        }
    }

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



```
    }  
  }  
}
```

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

```
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;  
            return 28;  
        }  
        return 31;  
    }  
}
```

```

    }
}
private static boolean isLeapYear(int year) {
    boolean Isleapyear = false;
    if (year % 400 == 0) {
        Isleapyear = true;
    }
    if((year % 4 == 0) && (year % 100 != 0)){
        Isleapyear = true;
    }
    return Isleapyear;
}
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;
            return 28;
        }
        return 31;
    }
}
}
}

```