Oppgaver uke 11 - Rekursjon

Oppgave 1

Bruk en lenkeliste når du løser deloppgavene 1a, 1b og 1c.

1a) Skriv ut elementene i lenkelista først i riktig rekkefølge, så i motsatt rekkefølge, ved bruk av rekursjon. Når du skriver ut i motsatt rekkefølge, skal du altså ikke bruke forrige-pekere, om lenkelisten din inneholder disse.

1b) Finn størrelsen på lenkelista ved bruk av rekursjon.

1c) Test at metodene du implementerte i 1a og 1b fungerer ved å skrive et lite testprogram.

Oppgave 2

2a) Vi har et vilkårlig antall kaniner. Alle kaninene har to ører som står rett til værs. Vi ønsker nå å skrive en rekursiv metode for å regne ut hvor mange ører kaninene har totalt. Metoden skal ta inn antall kaniner som argument.

```
class KaninOreTeller {
  //Teste Fifo-listen:
    public static void main(String[] args) {
        System.out.println(tellKaninOrer(0)); //Forventet resultat 0
        System.out.println(tellKaninOrer(1)); //Forventet resultat 2
        System.out.println(tellKaninOrer(2)); //Forventet resultat 4
        System.out.println(tellKaninOrer(12)); //Forventet resultat 24
        System.out.println(tellKaninOrer(234)); //Forventet resultat 486


    }

    public static int tellKaninorer(int antallKaniner) {
        //kode inn her
    }
}
```

2b) Noen ganger velger kaniner kun å ha ett øre opp. I dette tilfellet vil kaniner på partallsplasser (2,4,6, ...) ha to ører oppe, mens kaniner på oddetallsplasser (1,3,5, …) kun vil ha ett øre oppe. Endre metoden i **2a** slik at den tar hensyn til dette.

```
class KaninOreTeller {
    public static void main(String[] args) {
        System.out.println(tellKaninOrer(0)); //Forventet resultat 0
        System.out.println(tellKaninOrer(1)); //Forventet resultat 1
        System.out.println(tellKaninOrer(2)); //Forventet resultat 3
        System.out.println(tellKaninOrer(12)); //Forventet resultat 18
        System.out.println(tellKaninOrer(234)); //Forventet resultat 351
    }

    public static int tellKaninorer(int antallKaniner) {
        //din kode inn her
    }
}
```

Oppgave 3

3a) Skriv en metode som rekursivt finner antall forekomster av en string i en annen string. Du kan anta at ingen av strengene er tomme (""). (Et tips her kan være å bruke substring- og length-metodene til String)

```
class StringOprasjoner {
    public static void main(String[] args) {
       String str = "kattkukatt"
        System.out.println(antallSubStringer(str, "katt")); //Forventet resultat 2
        System.out.println(antallSubStringer(str, "ku")); //Forventet resultat 1
        System.out.println(antallSubStringer(str, "hund")); //Forventet resultat 0


    }

    public static int antallSubStringer(String str, String sub){
       //kode inn her
    }
}
```

3b) Lag en rekursiv metode som tar inn en string hvis en bokstav forekommer 2 ganger etter hverandre. F.eks. ved l i hello skal det plasseres en * i mellom de doble bokstavene (hello blir da hel*lo). (Et tips her kan være å bruke substring-, length- og charAt-metodene til String)

```
class StringOprasjoner {
    public static void main(String[] args) {
        System.out.println(settInnStjerner("hello")); //Forventet resultat hel*lo
        System.out.println(settInnStjerner("helllo")); //Forventet resultat hel*l*lo
        System.out.println(settInnStjerner("helloo")); //Forventet resultat hel*lo*o
        System.out.println(settInnStjerner("")); //Forventet resultat
        System.out.println(settInnStjerner("trollmannen")); //Forventet trol*lman*nen
    }

    public static String settInnStjerner(String str){
       //kode inn her
}
```

Oppgave 4

Oversett den iterative fakultet-metoden nedenfor til en rekursiv fakultet-metode. Beskriv hva som skjer fra du kaller på den rekursive metoden i main frem til den terminerer. La n = 5.

```
public static int factorialIterative(int n) {
     int result = 1;
     for (int i = 1; i <= n; i++) result *= i;
     return result;
}
```