Fakultet for ingeniør- og naturvitskap
Institutt for elektrofag

Høgskulen på Vestlandet

Obligatorisk innlevering 3 - ELE102 Programmering og mikrokontrollere

Programkodefiler skal inneholde et kommentarfelt i starten. Kommentarfeltet skal inneholde kode-forfatter sitt/sine navn. For GUI-prosjekter er det nok å ha forfatternavn i Form-klasse-filen.

For «ikke Arduino oppgaver»: du kan velge å implementere løsning som GUI eller CUI program.

Oppgave 1 – bruk av løkkestrukturer

En tekststreng kan være bygget opp av ord som er skilt med mellomrom, f. eks.:

```
string mangeOrd = "eple appelsin drue yoghurt kiwi";
```

Skriv et program som skiller ut ett og ett ord fra strengen **`mangeOrd`**, som skal defineres som vist over. Relevante ferdige funksjoner/property er **`IndexOf(…)`**, **`Substring()`** og **`Length`**. For hver iterasjon skal ett av ordene i **`mangeOrd`** lagres i en strengvariabel kalt **`etOrd`** og innholdet skal vises for brukeren. Alle ordene, med unntak av det siste, skal etterfølges av et komma og en ordskiller. Det endelige resultatet skal da se slik ut:

```
eple, appelsin, drue, yoghurt, kiwi
```

Tips:

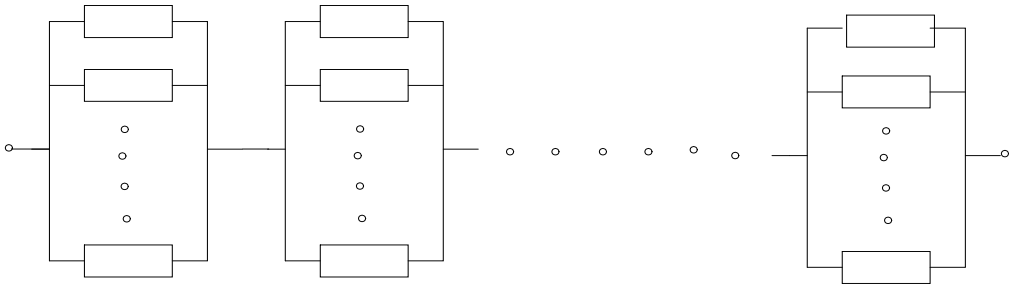
- *Bruk while-løkke.*
- *Husk at* **`IndexOf(…)`** *returnerer* **`-1`** *dersom den ikke finner det du søker etter.*

Fakultet for ingeniør- og naturvitskap
Institutt for elektrofag

Høgskulen på Vestlandet

Oppgave 2 – bruk av løkkestrukturer

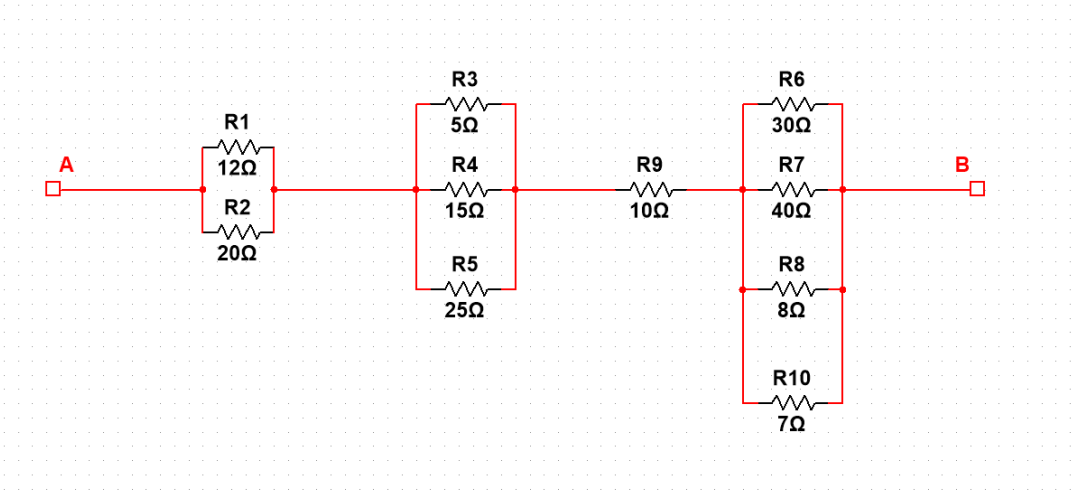
Vi ønsker et program som beregner totalmotstanden i et motstandsnettverk som består av en seriekopling av parallellkoplete motstander, slik figuren viser. Antall motstander i hver parallellkopling og antall parallellkoplinger er ikke kjent ved oppstart av programmet.



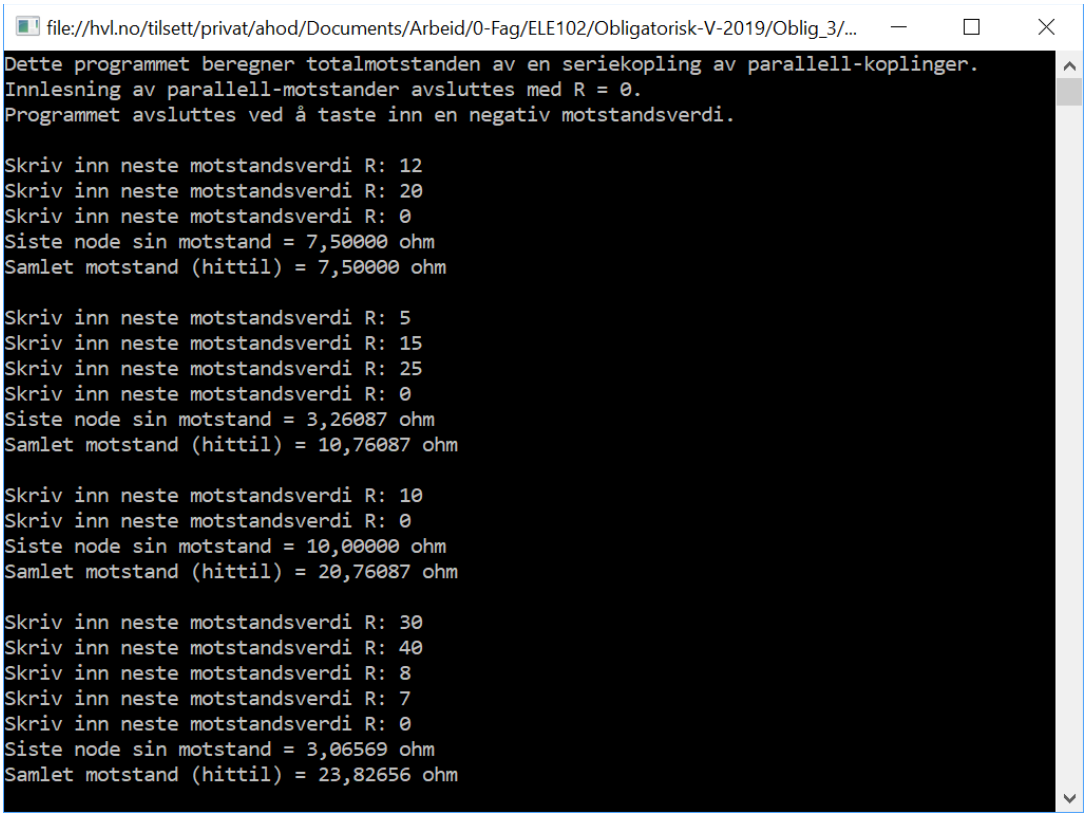
Programmet skal virke på følgende måte:

- Én motstandsverdi leses inn om gangen.
- Når innlest motstandsverdi = 0 betyr det at det er slutt på en parallellkopling – vi er kommet til en node.
- Dersom innlest verdi er < 0 betyr det at det er slutt på (parallellkopling og på) nettverket og totalmotstanden vises. NB: Programmet skal altså vise resultatet når bruker oppgir R < 0, men husk at bruker kanskje først oppgir R = 0 og deretter R < 0.

Eksempelkrets:



Eksempel - programutføring:



Tips:

- Beregninger utføres etter hvert som bruker taster inn verdier. Da trenger du ikke å lagre alle motstandsverdiene.
- Det er enklest å regne med konduktanser i parallellkoplingene og resistanser i seriekoplingene.
- Det er en fordel om programmet viser mellomresultater underveis. Da er det lettere å sjekke om det regner riktig.

Oppgave 3 – Arduino og millis

Når vi bruker `delay()`-funksjonen til å lage sanntidsklokker blir dette unøyaktig fordi syklustiden til loop() ikke bare avhenger av `delay(…)`, men også tiden det tar å utføre resten av koden i `loop()`.

For å oppnå bedre nøyaktighet i programmet vårt, skal vi i stedet for `delay` bruke funksjonen `millis`: (se lenken https://www.arduino.cc/reference/en/language/functions/time/millis/ for detaljer) Funksjonen returnerer antall millisekunder som er gått siden programmet har startet.

Modifiser koden fra oblig 1 - oppgave 5 slik at i stedet for `delay` programmet bruker funksjonen `millis`. Merk at et slik program vil gi oss mer presis klokke i forhold til det som bruker `delay` – finn ut hvorfor!

Oppgave 4 – Meldinger fra Arduino

Du skal lage et Arduino-program som overvåker status på to LED-lys (lys 1 som skal være grønt og lys 2 som skal være rødt). Lysene skal kunne slås på/av med to trykknapper. Programmet skal i tillegg lese verdien av en potensiometer som også er koblet til mikrokontrolleren.

Verdiene skal sendes som en melding via seriellporten. Meldingen skal bruke bokstaver A, B, C og D til å markere deler av meldingen og skal ha følgende format:

$AXX/XX/XXXXBXX:XX:XXCXXXXDXX#

der innholdet etter A er meldingsdato (DD/MM/ÅÅÅÅ), innholdet etter B er meldingstidspunkt (TT:MM:SS) innholdet etter C er verdien som leses av potentiometer (0000-1024) og innhold etter bokstaven D er status på to grønt/rødt-lys (1 for på , 0 for av): 00, 01 10 eller 11. Tegn $ markerer starten av en melding, tegnet # markerer slutten av en melding.

Eksempel på en slik melding når potensiometerverdi er 500, lys 1 (grønt) er av og lys 2 (rødt) er på:

$A30/02/2018B12:00:00C0500D01#

Meldingen skal i utgangspunktet sendes hvert sekund. Hint: programkode fra oppgave 3 vil kunne brukes til å «telle» tiden.