

Opgave 1

Der findes naturlige tal, hvor om det gælder, at kvadratet af tallet kan opdeles i to tal, hvis sum er lig det oprindelige tal (45 er et eksempel: 45 i anden potens er 2025, og 20 plus 25 er lig 45). Opgaven går ud på at skrive en metode, som kan udskrive samtlige disse tal, som er større end 1 og mindre end 100 på `System.out`. Du kan antage, at kvadrattallet af det oprindelige tal har et lige antal cifre.

Opgave 2

Der findes naturlige tal, hvor om det gælder, at summen af cifrene opløftet i deres positions potens er lig tallet. Fx er 135 et sådant tal, fordi summen af en i første potens, tre i anden potens og fem i tredje potens er lig tallet selv ($1+9+125=135$). Opgaven går ud på at skrive en metode, som kan udskrive samtlige disse tal, som er mindre end 1000 på `System.out`.

Opgave 3

Denne opgave går ud på at skrive en metode med følgende signatur:

```
boolean neighbors(String s)
```

Der returneres *true*, hvis parameteren *s* indeholder nabobogstaver i 'stigende' rækkefølge, fx opfylder *gh* kriteriet; men det gør *hg* ikke. Findes der ikke nabobogstaver returneres *false*. Opgaven skal løses for små bogstaver i det engelske alfabet (*abcdefghijklmnopqrstuvwxyz*).

Eksempler:

'hsgafkroditjsla' returnerer *false* – ingen nabobogstaver

'krstuoinnaqwlp' returnerer *true* – s og t er nabobogstaver

Opgave 4

Denne opgave går ud på at skrive en metode med følgende signatur:

```
boolean sumOfTwoEqualsThird(int arr[])
```

Parameteren *arr* er et array af heltal større end nul. Metoden returner *true*, hvis det forekommer, at summen af to elementer på forskellige pladser er lig med et tredje element. Hvis ikke returneres *false*.

Eksempler:

{17,29,512,58,43} returnerer *false*

{46,39,18,15,21} returnerer *true* ($18+21=39$)

{45,29,31,58,99} returnerer *false*; 29 må kun bruges én gang

Opgave 5

Et palindrom er et ord, som staves ens både forfra og bagfra. Eksempler: mellem, otto, kajak. Skriv en metode som kan afgøre om et ord er et palindrom. Signatur:

```
boolean isPalindrom(String text)
```

Sætninger kan også være palindromer, fx *en af dem der tit red med fane*. Udvid metoden således at den også kan håndtere sætninger.

Opgave 6

Det klassiske danske indregistreringsnummer til personbiler og visse andre køretøjer består af 7 karakterer. De to første er store bogstaver, og de fem sidste er tal, hvoraf de første ikke må være nul. Eksempler på valide indregistreringsnumre: MA39604 og KD47612. Skriv en metode som kan validere den type numre.

Opgave 7

Opgaven går ud på at skrive en metode, som kan afgøre om eventuelle parenteser i en tekststreng er matchede. Det vil sige, at der skal være lige mange start- og slutparenteser, og på et vilkårligt sted i teksten må der ikke have forekommet flere slut- end startparenteser.

Eksempler: `(())()` er ok; `()((((()))` er ok; `()())` er ikke ok; `)((())` er ikke ok. Løsningen skal tage højde for, at der kan forekomme andre tegn end start- og slutparenteser i teksten.

Opgave 8

Opgaven drejer sig om at få fastlagt hvor mange 6-cifrede tal der findes, hvorom det gælder, at ethvert ciffer er større end det foregående, fx 345678, 456789 og 256789.

Opgave 9

Du skal tilføje to metoder til *StringTokenizer*-klassen. Den første skal angive, hvor mange tegn/anslag teksten indeholder *uden* mellemrum/blanke, og den anden skal kunne finde ud af om teksten indeholder forekomster af to på hinanden følgende delimiters/skilletegn.

Opgave 10

Du skal skrive en metode, som har følgende signatur.

```
boolean sumOfTwoEqualsX(int arr[], int x)
```

Arrayet består af sorterede heltal > 0 , og metoden skal finde ud af om der findes to tal i arrayet hvis sum er lig med parameteren x . Hvert element i arrayet må kun tilgås/læses én gang.

På dette site findes nærmest en uendelig mængde af opgaver:

<https://www.w3resource.com/java-exercises/>