

Obligatoriske oppgave 01-A

Denne oppgaven omhandler tabeller og metoder og vil være en del av neste obligatorisk innlevering med frist i starten av uke 40.

Den obligatoriske innleveringen skal gjøres i grupper med opptil 4 personer. Det er de samme gruppene som dere skal bruke i programmeringsprosjektet i uke 41/42.

Husk å bruke startkoden og enhetstester for oppgaven:

<https://github.com/dat100hib/dat100-javainnlevering2/blob/master/README.md>

På siden ovenfor er det også beskrevet hvordan dere kan clone startkoden til egen PC og hvordan dere kan utføre enhetstestene.

Metodene i denne oppgaven skal implementeres fra bunden og det er derfor ikke lov å bruke metoder som finnes i Arrays-klassen.

bddfgøøøø

a)

Implementer en metode:

```
public static void skrivUt(int[] tabell)
```

som kan skrive ut en tabell med heltal. Du kan selv bestemme formatet.

b)

Implementer en metode:

```
public static String tilStreng(int[] tabell)
```

som kan returnere en tekststreng (String) med innholdet av en tabell. For en tabell med elementene 42,67,89 skal følgende streng returneres:

```
"[42,67,89]"
```

c)

Implementer en metode:

```
public static int summer(int[] tabell)
```

som gitt en **tabell** av heltall som parameter beregner summen av tallene som er lagret i tabellen.

Implementer tre varianter av metoden: en som bruker en vanlig for-løkke, en som bruker en while-løkke, og en som bruker en utvidet for-løkke.

Test de ulike implementasjoner av metoden på utvalgte tabeller og se at de gir riktig resultat. Metoden fra a) kan brukes ifm. test.

d)

Implementer en metode:

```
public static boolean finnesTall(int[] tabell, int tall)
```

som gitt en **tabell** og et **tall** returnerer true om tabellen **tabell** inneholder tallet **tall** og false ellers.

e)

Implementer en metode:

```
public static int posisjonTall(int[] tabell, int tall)
```

som returnerer det første index (posisjon) i **tabell** som inneholder tallet **tall** og -1 ellers.

f)

Implementer en metode:

```
public static int[] reverser(int[] tabell)
```

som gitt en **tabell** av heltall returnerer en ny tabell som har de samme elementene som **tabell** men i modsatt rekkefølge.

g)

Implementer en metode:

```
public static boolean erSortert(int[] tabell)
```

som kan avgjere om ein array (tabell) av heiltal er sortert stigande (dvs. at alle tal skal vere større enn det forrige)

h)

Implementer en metode:

```
public static int[] settSammen(int[] tabell1, int[] tabell2)
```

som gitt to tabeller `tabell1` og `tabell2` som parametre og returnerer en ny tabell som er sammensetning av de to tabeller.