

Obligatorisk oppgave nr.4

PROG1001 - Grunnleggende programmering, høsten 2025

Innleveringsfrist: 13.november 2025 kl.11:00 (må overholdes)
i Blackboard på pdf-format
(Code::Blocks: File → Export → As PDF (*uten linjenumre!*))

Du skal bruke/ta utgangspunkt i `oblig4.tpl`

(som du selv omdørper til `oblig4.c`)

Bruk gjerne også `LesData.h` som introduseres i `EKS_34.c`.

Oppgaven går ut på at man skal registrere produsenter og deres produkter. Man skal kunne legge til produsenterprodukter og skrive alt dette ut på ulike formater.

Av vedlegget kan man se at det skal lages innmat til følgende 13 funksjoner:

```
struct Produsent* finnProdusent(const char* navn)
    Returnerer om mulig (ellers NULL) en peker til produsenten som har navn.
struct Produkt* finnProdukt(const struct Produsent* produsent,
                           const char* navn)
    Returnerer om mulig (ellers NULL) en peker til produktet med navn hos produsent.
```

```
void nyProdusent()
```

Det kommer en egen melding om det allerede er fullt med produsenter. I motsatt fall leses en produsents navn. Finnes denne fra før, kommer det også en melding. Ellers lages en ny produsent, antall produsenter telles opp, og produsentens data leses inn av:

```
void produsentLesData(struct Produsent* produsent, const char* navn)
    Produsentens navn lagres unna, byen leses og antall produkter nullstilles. Deretter opprettes og leses det inn produkter så lenge brukeren ønsker. Det kommer egne meldinger om produktet allerede finnes hos produsenten (jfr. finnProdukt(...)), eller at det blir fullt med produkter. Ett produkt hos en produsent legges inn av:
```

```
void produsentNyttProdukt(struct Produsent* produsent,
                           const char* navn);
```

Legger inn ett nytt produkt med navnet navn hos produsent.

Allt produktets data leses inn/registeres av:

```
void produktLesData(struct Produkt* produkt, const char* navn)
    Produktets navn lagres unna, og de to andre datamedlemmene leses inn.
```

```
void leggTilEttProdukt()
```

Det kommer en egen melding om ingen produsenter finnes. I motsatt fall leses en produsents navn. Finnes ikke denne, kommer det en egen melding. Ellers leses et produktnavn. Finnes denne allerede hos produsenten, kommer det også en egen melding. Ellers registreres dette ene nye produktet hos produsenten vha.
produsentNyttProdukt(...).

```
void skrivAbsoluttAlt()
```

Det kommer en egen melding om ingen produsenter finnes. I motsatt fall skrives *alle* data om *alle* produsentene ut på skjermen, inkludert deres nr (1 til gAntallProdusenter), samt *alt* om *alle* produkter (også nummeret) hos hver enkelt produsent.

Til dette brukes funksjonene:

```
void produsentSkrivData(const struct Produsent* produsent)
void produktSkrivData(const struct Produkt* produkt)
```

```
void fjernAllkokerteData()
```

Funksjonen går igjennom *alle* produsentene, og frigir *alt* allokkert minne/memory hos *alle* produsentene og *alle* deres produkter. Til dette brukes bl.a:

```
void produsentSlettData(struct Produsent* produsent)
void produktSlettData(struct Produkt* produkt)
```

NB: Husk god og *korrekt* (ellers risikerer den å ikke bli godkjent) kommentering, innrykk, leseighet og at linjene ikke blir for lange (slik at teksten wrapper ved en evt. utskrift)

Lykke til!

Martin Kristensen Eide, Peter Behncke Nes & FrodeH