

Obligatorisk oppgave nr.1

PROG1003 – Objekt-orientert programmering, våren 2026

Innleveringsfrist: 5.februar 2026 kl.11:00 (må overholdes)
i Blackboard på pdf-format

Følgende data/variable *skal* brukes i programmet:

```
const int ANTSTOPP = 11;          ///< Totalt antall ulike busstopp.

struct Rute {
    vector <string> stopp;         // Rutens ulike stoppesteder.
    int ruteNr,                   // Reelt rutenr, f.eks. 42, 165, 718
        totMin;                   // Totalt antall minutter å kjøre på ruten
};                                // (fra første til siste stoppested).

vector <Rute*> gRuter;              ///< Pekere til rutene.

const vector <string> gBusstopp =  ///< Navn på alle busstopp.
{ "Skysstasjonen", "Fahlstrøms plass", "Sykehuset",
  "Gjøvik stadion", "Bergslia", "Overby", "Nybrua",
  "NTNU", "Kallerud", "Hunndalen", "Mustad fabrikker" };

const int gMinutter[ANTSTOPP][ANTSTOPP] = ///< Min.mellom stoppesteder.
// const vector <vector <int> > gMinutter = // Alternativt (nå lært i vår).
{ { 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, // Skysstasjonen = 0
  { 3, 0, 3, 0, 0, 0, 3, 0, 0, 0, 4 }, // Fahlstrøms plass = 1
  { 0, 3, 0, 1, 0, 0, 0, 0, 0, 0, 0 }, // Sykehuset = 2
  { 0, 0, 1, 0, 3, 0, 0, 0, 0, 0, 0 }, // Gjøvik stadion = 3
  { 0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0 }, // Bergslia = 4
  { 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0 }, // Øverby = 5
  { 0, 3, 0, 0, 0, 0, 0, 2, 0, 0, 2 }, // Nybrua = 6
  { 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 0 }, // NTNU = 7
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, // Kallerud = 8
  { 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 2 }, // Hunndalen = 9
  { 0, 4, 0, 0, 0, 0, 2, 0, 0, 2, 0 } }; // Mustad fabrikker = 10
```

Vi ser at det f.eks. er 4 min mellom Fahlstrøms plass og Mustad fabrikker (selvsagt begge veier), og at det er 2 min mellom Bergslia og Øverby (også begge veier). '0' betyr at det *ikke* er noen direkte forbindelse kun mellom disse stoppestedene.

Følgende 10 funksjoner *skal* lages (i tillegg til et komplett program og `main` som styrer alt):

`void skrivMeny();`

Lovlige kommandoer er (i tillegg til `Quit/avslutt`):

N - Ny rute, S - Slett rute, A - skriv Alle ruter og B - skriv alle Busstopp

void skrivStopp()

Skriver navnet på *alle* stoppestedene (som er i `gBusstopp`), nummerert fra 1 og oppover.

void skrivRuter()

Skriver ut *alle* data for *alle* rutene (med '-->' mellom stoppestedene, unntatt etter den siste) ved bl.a. å bruke funksjonen **void ruteSkrivData(const Rute rute)**

Det skrives også ut rutenes indeks, fra 1 og oppover (i tillegg til deres reelle `ruteNr`).

void skrivNesteStoppesteder(const int stopp)

Skriver *alle* aktuelle/lovlige stoppesteder etter/ut fra stoppested nr. `stopp`.

void nyRute()

Oppretter en ny rute. Får *alle* dennes data lest inn vha. *funksjonen nedenfor*. Gikk innlesningen bra, så legges den inn *bakerst* i `gRuter`, og dens indeks (husk vi nummererer fra 1) skrives ut sammen med *alle* dens data (jfr. **ruteSkrivData(...)**), ellers slettes ruten.

bool ruteLesData(Rute & rute)

Leser rutens *reelle* nummer (trenger *ikke* å sjekke at andre allerede har dette rutenummeret). *Alle* mulige stoppesteder skrives ut på skjermen. Det første som leses blir da rutens startsted. Leser så indeksen for neste *lovlige* stoppested (bruk tidligere nevnt funksjon), ut fra nåværende stoppested, inntil brukeren velger '0' (null). Navnet på stoppestedene legges kontinuerlig til *bakerst* i rutens `stopp`. Dens `totMin` oppdateres også hele tiden. Er det *mer enn ett* stoppested på ruten, returneres `true` ellers *false*. Du trenger *ikke* å sjekke for duplikate forekomster av stoppesteder.

void slettRute()

Er det tomt for ruter, kommer en egen melding. I motsatt fall skrives *alle* rutene ut (inkludert deres indekser fra 1 og oppover). Brukeren spørres så om hvilke(n) som skal slettes.

-1 (minus 1) betyr *alle*, og da kalles **void slett()** som gjør dette.

0 (null) betyr ingen, og det kommer en egen melding, før funksjonen avsluttes.

1 og oppover betyr at den aktuelle ruten slettes. Til dette brukes funksjonen

void slett(const int nr) som flytter den *bakerste* i `gRuter` til den slettedes plass (indeks).

NB1: Husk at i både `gRuter` og `gMinutter` så ligger alt f.o.m. indeks nr.0 (null) og oppover, selv om vi i *hele* programmet for dette bruker/skriver ut indeks 1 (en) og oppover.

Bruk `LesData2.h`

NB2: Husk å skrive *ren C++-kode* (og ikke C, med bl.a. `printf`, `scanf`, `gets`, `malloc` og `free`), god og *korrekt* kommentering, innrykk, leselighet og at linjene ikke blir for lange (så teksten wrapper ved en evt. utskrift) - ellers risikerer den dessverre å ikke bli godkjent.

Tips: `EKS_06.CPP`

Lykke til!

Martin Kristensen Eide, Peter Behncke Nes og FrodeH