NTNU
Kunnskap for en bedre verden

Institutt for datateknologi og informatikk

Eksamensoppgave i PROG1003
– Objekt-orientert programmering

Faglig kontakt under eksamen:	**Frode Haug**
Tlf:	**950 55 636**
Eksamensdato:	**18.mai 2020** **- HJEMME**
Eksamenstid (fra-til):	**09:00-13:00 (4 timer)**
Hjelpemiddelkode/Tillatte hjelpemidler:	**F - Alle trykte og skrevne.** (kalkulator er *ikke* tillatt)
Annen informasjon:	
Målform/språk:	**Bokmål**
Antall sider (inkl. forside):	**7**

Informasjon om trykking av eksamensoppgaven

Originalen er:

1-sidig X **2-sidig** ☐

sort/hvit X **farger** ☐

Skal ha flervalgskjema ☐

Kontrollert av:

_____

Dato Sign

NB: Oppgave 1a, 1b, 1c og 2 er totalt uavhengige og kan derfor løses separat.

Oppgave 1 (30%)

I hele oppgave 1, være nøye med å :

- skrive *hele* funksjonen, dvs. *både* headingen og *hele* innmaten
- bruke `const` og/eller '&' (når dette er aktuelt) ifm. parametre
- kommentere funksjonen (hva den gjør, evt. parametre og evt. returverdi) etter Doxygen

a)

Lag funksjonen `int antallAvToBokstaver(.....)`

Funksjonen tar en tekst (`string`) og to tegn (`char`) som parametre.
Den returnerer *totalantallet* av hvor mange det *til sammen* er av de to tegnene i teksten.

Eks: Parameter-teksten: «arendal, arsenal, arne, arnesen, arvika, arken, arbeiderbladet»
Bokstavene 'a' og 'r' forekommer totalt **19** ganger (altså det funksjonen returnerer)

b)

Lag funksjonen `int oppdaterOgSummer(.....)`

Funksjonen tar *kun* en `vector` med `int`'er som parameter. Den går gjennom *den originale vectoren*, og oppdaterer alle elementene/«skuffene» med *oddetalls indeks* til å bli det *tredobbelte i verdi*. Den returnerer *totalsummen* av alle disse oppdaterte verdiene.

Eks: vectoren før: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
vectoren etter: 0 **3** 2 **9** 4 **15** 6 **21** 8 **27** 10 **33** 12 **39** 14 **45** 16 **51** 18 **57**
Summen av oddetall-indeks tallene: **300** (altså det funksjonen returnerer, i tillegg til at parameter-vectoren er endret)

c)

Lag funksjonen `void endreVerdier(.....)`

Funksjonen tar to parametre:

- en `map` med `int`-keyer og tilknyttede tekster (`string`)
- en `string` 's'

Funksjonen går manuelt, vha. iterator, gjennom *hele* den *originale* `map`'en, og *alle* `int`-keyen som er modbare med 3 (dvs. key % 3 = 0) får sin tilknyttede tekst erstattet med '`s`'.

Lag også en versjon av innmaten, der *alt* gjøres *kun* vha. *en* `for_each` -funksjon, bl.a med en lambda-funksjon som parameter.

Oppgave 2 (70%)

Les *hele* teksten for denne oppgaven (2a-2g) *nøye*, før du begynner å besvare noe som helst. Studér vedlegget, som inneholder mange viktige opplysninger som du trenger/skal bruke. Legg spesielt merke til `constene`, klassen med datamedlemmer og (ferdiglagde) funksjoner, globale variable, `main` og `skrivMeny()`.
Husk også på de ferdiglagde funksjonene på `LesData2.h`. <u>Bruk *alt* dette *svært* aktivt.</u>

Vi ser for oss det totalt utopiske, utenkelige og utrolige at en hel nasjon er utsatt for en pandemi, og at nasjonens husstander (med kanskje flere personer i hver) *kan* få karantene et visst antall dager. Vi skal lage et program som vha. en liste med husstander holder orden på alt dette.

Datastrukturen

Datastrukturen består kun av *listen* `gHusstander`, samt `gDagnummerIAret` (se vedlegget). I listen *kan* og vil det ofte forekomme mange objekter med samme ID (`dagSlutt`). Men, ingen kontaktperson (`navn`) for en `Husstand` har samme navn (selv om vi ikke har noen direkte sjekk på/av dette noe sted i selve koden som skal lages/skrives).

Oppgaven

a)

Skriv innmaten til `void skrivAlleHusstander()` **og**
`void Husstand::skrivData()`

Den første funksjonen skriver først ut dagens absolutte nummer i året. Helt til slutt skriver den ut antall husstander i listen. Mellom dette skriver den (sammen med den andre funksjonen) ut *alle* data om *alle* husstander på skjermen. For hver husstand skrives også hvilket nummer (1, 2, 3,….) den er i listen.

b)

Skriv innmaten til `void nyHusstand()` **og**
`void Husstand::lesData()`

Den første funksjonen oppretter en ny `Husstand`, lar denne selv (vha. den andre funksjonen) lese inn *alle* sine data, legger den inn i listen, og sørger for at denne forblir sortert på ID/`dagSlutt`. **NB:** `dagSlutt` leses *ikke* inn direkte, men bli beregnet ut fra `antallDager` og `gDagnummerIAret`.

c)

Skriv innmaten til `void endreHusstand()` **og**
`void Husstand::endreData()`

Den første funksjonen kommer med en melding om det ikke finnes noen husstander. I motsatt fall leser den inn et *nummer for en husstand i listen* (fra 0 (null) til antall i listen). '0' betyr at brukeren angrer seg, og resten av funksjonens innmat hoppes over, men det kommer en melding om at ingenting skjer. Ellers finner funksjonen frem til den aktuelle husstanden (ut fra dets nummer i listen), og lar brukeren endre på (vha. den andre funksjonen) husstandens antall karantenedager som er igjen. Husk at listen etterpå fortsatt skal være sortert.
Den andre funksjonen skriver både helt først og til slutt ut *alle* husstandens data. Mellom dette skriver den ut antall dager husstanden har vært i karantene, hvilket absolutt dagnummer i året karantenen startet og hvor mange dager som er igjen. Den lar også brukeren *endre på antall dager som er igjen* (dermed må `dagSlutt` og `antallDager` regnes ut på nytt).
For enkelhets skyld skal antall dager som er igjen være mellom 1 (en) og `MAXKARANTENE` *totalt* (dvs. øvre grense er `MAXKARANTENE` *minu*s antall dager *hittil* i karantene).

d) **Skriv innmaten til** `void fjernHusstander()` **og**
`void Husstand::skrivTilFil(ofstream & ut)`

Legg merke til i `main` hvor den første funksjonen kalles. Funksjonen fjerner *alle* husstander (både objekt og peker til den) i listen som har `sluttdag` lik `gDagunmmerIAret`. Disse ligger evt. *alltid* aller først i listen! Men, *før* en `Husstand` slettes helt, skrives alle dens data vha. den andre funksjonen, *bakerst (append) på filen* «KARANTENEFERDIG.DTA» - på et selvvalgt format. *Om* noen ble fjernet, skriver den til slutt ut dette antallet, og at de er *skrevet/ lagt til bakerst* på nevnte fil. **NB:** Husk at listen også kan bli helt tom under denne operasjonen.

e) **Skriv innmaten til** `void statistikk()`

Funksjonen går i to for-løkker (begge med en `'i'` som går fra 1 til `MAXKARANTENE`) etter hverandre, og disse to for-løkkene skriver ut statistikk/oversikt over henholdsvis:

1) Antall husstander ute av karantene om `'i'` dager ift. `gDagnummerIAret`
2) Antall husstander *og personer* med `'i'` dagers karantene (deres `antallDager`).

Det skal *ikke* komme utskrift om antall husstander for en `'i'` er lik 0 (null).

f) **Skriv innmaten til** `void slettPersonsHusstand()`

Den kommer en melding om det ikke finnes noen husstander. I motsatt fall leser den inn et ønsket navn fra brukeren. *Om* ingen husstand har noen kontaktperson med dette navnet, kommer også en melding. I motsatt fall skrives alle husstandens data ut på skjermen, før objektet slettes fra både hukommelsen og lista. **NB:** De som slettes skal *ikke* skrives bakerst på filen i oppg.2d.

g) **Skriv innmaten til** `void lesFraFil()` **og**
`Husstand::Husstand(ifstream & inn)`

Disse funksjonene sørger til sammen for at *alle* data om *alle* husstandene leses inn fra filen «HUSSTANDER.DTA». Aller først på filen ligger `gDagnummerIAret` og antall husstander som videre kommer på filen. Formatet for disse `Husstand`-postene bestemmer du selv, men **dette *skal* oppgis som en del av besvarelsen**. Husk at listen til slutt skal være sortert.

Annet (klargjørende):

- Vi forutsetter at:
  - programmet *ikke* brukes over et årsskifte, slik at `gDagnummerIAret` *alltid* er gyldig.
  - `gDagnummerIAret` hele tiden blir korrekt ved lesing fra fil, dvs. at den *alltid* er dagen i dag sitt reelle absolutte nummer i året. Dette sikres ved at programmet *startes/brukes hver eneste dag*, samt at det *alltid* avsluttes (og skriver til fil) ved dagens slutt.
- En husstands karantenetid er i utgangspunktet i intervallet fra 5 til 21 (jfr. `const`'ene og oppg.2b). Men, i oppg.2c *kan* denne endres med helt ned til bare *en* dag. Dette er helt greit – ellers hadde beregningen av minimumsgrensen i oppg.2c blitt vel småtuklete. Dette er også grunnen til at for-løkkene i oppg.2e går fra 1 (en) og oppover.
- Det er *ikke* en del av denne eksamensoppgaven å lage kode som *skriver hele listen til fil*.
- Du *skal* bruke `LesData2.h` ifm. løsningen av denne oppgaven. Du får nok også bruk for (deler av) pensumets temaer innen STL, men *ikke* bruk templates eller stoff/biblioteker utenfor pensum.
- Gjør dine egne forutsetninger og presiseringer av oppgaven, dersom du skulle finne dette nødvendig. Gjør i så fall klart rede for disse *i starten* av din besvarelse av oppgaven.

Lykke til på tidenes første (Corona-befengte) eksamen i PROG1003!

FrodeH

Vedlegg til PROG1003, 18.mai 2020: Halvferdig programkode

```
#include <iostream>                 //  cout, cin
#include <fstream>                  //  ifstream, ofstream
#include <string>                   //  string
#include <list>                     //  list
#include <algorithm>                //  Du får sikkert bruk for noen av disse ……
#include "LesData2.h"               //  Verktøykasse for lesing av diverse data
using namespace std;


const int MINKARANTENE =  5;        ///<  Min. antall karantenedager.
const int MAXKARANTENE = 21;        ///<  Max. antall karantenedager.
const int MAXIHUSSTAND = 10;        ///<  Max. antall personer i EN husstand.


/**
 *  Husstand (med hvilke absolutt dagnummer i året karantenen er slutt
 *            (= ID, listen er sortert på dette), antall dager i karantene, antall
 *           personer i husstanden, mobilnummer/navn/adresse til kontaktpersonen).
 */
class Husstand  {
  private:
    int    dagSlutt,                //  = ID - er sortert på dette.
           antallDager,
           antallPersoner,
           mobilNr;
    string navn,
           adresse;

  public:
    Husstand()  { dagSlutt = antallDager = antallPersoner = mobilNr = 0; }
    Husstand(ifstream & inn);                                          //  Oppgave 2G
    void   endreData();                                                //  Oppgave 2C
    int    hentAntall() const  {  return antallPersoner;  }
    int    hentDager()  const  {  return antallDager;     }
    int    hentID()     const  {  return dagSlutt;        }
    string hentNavn()   const  {  return navn;           }
    void   lesData();                                                  //  Oppgave 2B
    void   skrivData() const;                                          //  Oppgave 2A
    void   skrivTilFil(ofstream & ut) const;                           //  Oppgave 2D
};


void endreHusstand();                //  Oppgave 2C
void fjernHusstander();              //  Oppgave 2D
void lesFraFil();                    //  Oppgave 2G
void nyHusstand();                   //  Oppgave 2B
void skrivAlleHusstander();          //  Oppgave 2A
void skrivMeny();
void slettPersonsHusstand();         //  Oppgave 2F
void statistikk();                   //  Oppgave 2E


int  gDagnummerIAret;                ///<  Absolutt dagnummer i året (1-344).
                                     //      (344 + MAXKARANTENE = 365)
list <Husstand*> gHusstander;        ///<  ALLE husstandene i karantene.


int main() {
    char valg;

    lesFraFil();                                    //  Oppgave 2G

    fjernHusstander();                              //  Oppgave 2D
```

```
    cout << "\nI dag er det den " << gDagnummerIAret << ".dagen i året.\n\n";

    skrivMeny();
    valg = lesChar("\n\nKommando");

    while (valg != 'Q') {
        switch (valg) {
          case 'A': skrivAlleHusstander();   break;  //  Oppgave 2A
          case 'N': nyHusstand();            break;  //  Oppgave 2B
          case 'E': endreHusstand();         break;  //  Oppgave 2C
          case 'S': statistikk();            break;  //  Oppgave 2E
          case 'P': slettPersonsHusstand();  break;  //  Oppgave 2F
          default:  skrivMeny();             break;
        }
        valg = lesChar("\n\nKommando");
    }

    cout << "\n\n";
    return 0;
}



// ------------------------------------------------------------------------
//                       DEFINISJON AV KLASSE-FUNKSJONER:
// ------------------------------------------------------------------------

/**
 *  Oppgave 2G - Leser inn ALLE egne data fra fil.
 *
 *  @param  inn  -  Filobjektet det leses inn data fra
 */
Husstand::Husstand(ifstream & inn)  {                /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2C - Endrer karantenetiden for hele husstanden.
 */
void Husstand::endreData()  {                        /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2B - Leser inn ALLE egne data fra tastaturet.
 */
void Husstand::lesData()  {                          /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2A - Skriver ALLE egne data ut på skjermen.
 */
void Husstand::skrivData() const  {                  /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2D - Skriver ALLE egne data ut på fil.
 *
 *  @param   ut  -  Filobjektet det skrives ut data til
 */
void Husstand::skrivTilFil(ofstream & ut) const  {   /*   LAG INNMATEN  */   }
```

```
// ------------------------------------------------------------------------
//                        DEFINISJON AV ANDRE FUNKSJONER:
// ------------------------------------------------------------------------

/**
 *  Oppgave 2C - Endrer (om mulig) en nummerangitt husstand
 *               sine antall karantenedager.
 */
void endreHusstand()  {                                 /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2D - Fjerner ALLE husstander hvis karantene utgår I DAG.
 */
void fjernHusstander()  {                               /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2G - Leser ALLE husstandene i karantene inn fra fil.
 */
void lesFraFil()  {                                     /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2B - Legger inn en ny husstand i karantene.
 */
void nyHusstand()  {                                    /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2A - Går gjennom og skriver HELE 'gHusstander' sitt innhold.
 */
void skrivAlleHusstander()  {                           /*   LAG INNMATEN  */   }


/**
 *  Skriver programmets menyvalg/muligheter på skjermen.
 */
void skrivMeny()  {
    cout << "\nFølgende kommandoer er tilgjengelige:\n"
         << "   A  - skriv Alle husstander\n"
         << "   N  - Ny husstand i karantene\n"
         << "   E  - Endre karantenedagene for en husstand\n"
         << "   S  - Statistikk om alle husstander i karantene\n"
         << "   P  - slett en navngitt Persons husstand\n"
         << "   Q  - Quit / avslutt\n";
}


/**
 *  Oppgave 2F - Sletter (om mulig) en navngitt kontaktpersons sin husstand.
 */
void slettPersonsHusstand()  {                          /*   LAG INNMATEN  */   }


/**
 *  Oppgave 2E - Skriver statistikk over:
 *                1) Antall husstander ute av karantene om X (1-21) dager
 *                2) Antall personer/husstander med Y (1-21) dagers karantene.
 */
void statistikk()  {                                    /*   LAG INNMATEN  */   }
```