



Høgskolen i Gjøvik

Avdeling for Teknologi

E K S A M E N

FAGNAVN: Grunnleggende datakunnskap og programmering

FAGNUMMER: L 182 A

EKSAMENS DATO: 5. desember 2001

KLASSE: 01HIND*, 01HINE*, 01HDMU*, 01HING*

TID: 09.00-13.00

FAGLÆRER: Frode Haug

ANTALL SIDER UTLEVERT: 7 (inkludert denne forside)

TILLATTE HJELPEMIDLER: Kun læreboka "OOP in C++"

- Kontroller at alle oppgavearkene er tilstede.
- Innføring med penn, evt. trykkblyant som gir gjennomslag.
Pass på at du ikke skriver på mer enn ett innføringsark om gangen (da det blir uleselige gjennomslag om flere ark ligger oppå hverandre når du skriver).
- Ved innlevering skilles hvit og gul besvarelse og legges i hvert sitt omslag.
Oppgavetekst, kladd og blå kopi beholder kandidaten.
- Ikke skriv noe av din besvarelse på oppgavearkene.
- Husk kandidatnummer på alle ark.

Dette eksamenssettet består av tre ulike oppgavetyper:

Oppgave 1 omhandler teori fra datateknikk-delen.

Oppgave 2 omhandler hva som blir utskriften fra noen ulike program.

Oppgave 3 omhandler et litt større programmerings-case.

NB: De tre oppgavene er totalt uavhengige og kan derfor løses separat.

Oppgave 1 (30 %)

a) Internet:

a1) Nevn fem kategorier for tradisjonelle tjenester på Internet.

a2) Fortell kort (max. ½ A4-ark) omkring adressering på Internet (IP-adresse, nodenavn, domenenavn, (sub)domene og DNS).

a3) Forklar kort (max. ½ A4-ark) hva "Listserv" er.

b) Datamaskinen:

b1) Tegn et generelt blokkskjema for en datamaskin.

b2) Redegjør kort om (max. ½ A4-ark pr. punkt):

b2-a) USB

b2-b) Bussen

b2-c) CD-ROM

c) Datasikkerhet:

c1) Nevn fem områder/typer for datakriminalitet.

c2) Nevn fem eksempler på "fysiske sikringstiltak".

c3) Innen kryptering har vi bl.a. begrepene "integritet" og "autentisering". Forklar/definer disse.

d) Personvern / etikk:

d1) Hvilke fem hovedoppgaver har Datatilsynet ?

d2) Hva sier personopplysningsloven om video-/fjernsynsovervåkning ?

d3) Hva er forskjellen på "etikk", "moral" og "etiske retningslinjer" ?

Oppgave 2 (20 %)

a) Hva blir utskriften fra følgende program (litt hjelp: det blir 5 linjer):

```
#include <iostream>

using namespace std;

char txt[] = "STORT STORHAMAR";
int indeks[] = { 7, 12, 2, 3, 8, 9 };

int main() {
    int i;
    for (i = 0; i <= 12; i+=5)
        cout << txt[i+1] << '\n';

    for (i = 1; i < 3; i++)
        cout << txt[indeks[i]] << '\n';

    return 0;
}
```

b) Hva blir utskriften fra følgende program (litt hjelp: det blir 5 linjer):

```
#include <iostream>
#include <cstring>

using namespace std;

void funkl(char t[], int nn = 4) {
    cout << t[nn] << '\n';
}

void funk2(int & nn, char t[]) {
    nn = strlen(t) / 2;
}

int funk3(int nn, int mm) {
    int s = 1;
    for (int i = 1; i <= mm; i++) s *= nn;
    return s;
}

int main() {
    int n;
    funkl("LUNGEMOS");
    funkl("STÅLTERMOS", 9);
    funk2(n, "PØLSE, LOMPE OG BRØD"); cout << n << '\n';
    cout << funk3(2, 5) << '\n';
    funk2(n, "TØRRFISK"); cout << funk3(3, n) << '\n';
    return 0;
}
```

Oppgave 3 (50 %)

NB: Les hele teksten for denne oppgaven nøye, før du begynner å besvare noe som helst. Studer vedlegget nøye, som inneholder mange viktige opplysninger som du trenger/ skal bruke. Legg spesielt merke til (og bruk) de tre overloadede funksjonene “les”.

Innledning

I denne oppgaven skal du lage et lite program som holder orden på et arkiv/kartotek av CD'er. For hver CD skal det lagres: navnet til artisten/gruppen, tittel, utgivelsesår, antall sanger/låter og tittelen på hver enkelt sang. Alle dataene om CD'ene ligger/lagres på en egen fil. Din oppgave blir her å skrive resten av programmet angitt i vedlegget og denne oppgaveteksten.

Datastrukturen

I vedlegget kan du på den første siden se deklarasjoner/definisjoner av const'er, en struct og globale variable (en int og en struct-array). Dette skal være alt det du trenger av slike globale variable for å løse denne eksamensoppgaven !

Datastrukturen vil derfor bestå av arrayen “cder” som er bygd opp av “CD”-struct'er. Indeksene fra 1 til “ant_cder” er i bruk av denne arrayen. Av vedlegget kan du se hva hver enkelt struct inneholder.

Legg spesielt merke til den to-dimensjonale char-arrayen “sangtittel”. Hver linje i denne inneholder tittelen på en av CD'ens sanger. Vi bruker indeksene f.o.m. nr.0 t.o.m. “ant_sanger”-1.

Under løsningen av denne eksamensoppgaven skal du bruke denne datastrukturen !

Oppgaven

a) Lag funksjonen “void ny()”.

Funksjonen sjekker først om det er plass til flere CD'er i kartoteket. Om så ikke er tilfelle, kommer det en melding om det. I motsatt fall så tas en ny indeks i bruk av arrayen “cder” (ved å telle opp “ant_cder”). Deretter leses alle dataene inn om CD'en (vha. “les”-funksjoner). Utgivelsesår *skal* være et tall i intervallet 1950-2010. Antall sanger *skal* være et tall mellom 3 og “MAX_SANGER”.

b) Lag funksjonen “void oversikt()”.

Funksjonen går gjennom og skriver ut data om alle CD'ene i kartoteket. For hver CD skal det skrives to linjer på skjermen. Den første linjen inneholder CD'ens nummer i kartoteket, artistens/ gruppens navn og til slutt utgivelsesår i parentes. Den andre linjen inneholder tittel og antall sanger på CD. Utskriften skal stanse for hver tiende CD og vente på et tastetrykk fra brukeren. **NB:** Sangtitlene skal *ikke* skrives ut her ! De vil evt. bli skrevet ut ifm. oppgave e) nedenfor.

c) Lag funksjonen “void fjern()”.

Funksjonen ber brukeren om et nummer mellom 0 og “ant_cder”. Om nummeret er 0, så betyr det at brukeren angrer på fjerningen, og intet skjer. Er nummeret derimot større enn 0, så fjernes/ slettes den aktuelle CD'en fra kartoteket. Den siste CD'en flyttes inn i ‘hullet’ som oppstår, og “ant_cder” oppdateres korrekt.

d) Lag funksjonen “void les_fra_fil()”.

Dataene om alle CD’ene ligger lagret på filen “CD_INFO.DTA”. For hver post (CD) har denne følgende format: Først ligger utgivelsesåret etterfulgt av *ett* blankt tegn. På resten av den første linjen ligger artistens/gruppens navn. Alene, på den andre linjen ligger tittel. På den tredje linjen ligger kun et tall som angir antall sanger på CD’en. Så kommer det “ant_sanger” linjer, der hver inneholder en sangtittel. Funksjonen skal lese en fil med dette formatet inn i datastrukturen.

Husk å oppdatere variabelen “ant_cder”, og pass på at vi ikke ‘sprenger’ “cder”-arrayen.

Husk også på at artist-/gruppenavn, titler og sangtitler alle kan inneholde flere ord.

e) Lag funksjonen “void skriv()”.

Funksjonen skal be om og lese et artist-/gruppenavn (bruk relevant hjelpefunksjon til dette).

Deretter leser den brukerens ønske, som *skal* være ett av tegnene/bokstavene ‘u’, ‘U’, ‘m’ eller ‘M’. Funksjonen går så gjennom hele kartoteket. For hver CD som har “navn” lik det brukeren skrev inn, så skrives *en* linje med CD’ens tittel, utgivelsesår og antall sanger. *Om* brukeren ovenfor tastet inn ‘m’ eller ‘M’ (der ‘M’ står for ‘Med’), så skrives også alle sangtitlene på CD’en ut på hver sin linje. I dette tilfellet må brukeren trykke en tast mellom hver utskrift av CD’er.

(Om brukeren har valgt ‘u’ eller ‘U’ (som står for ‘Uten’), så trengs aldri en slik tastetrykking, selv om antall CD’er fra vedkommende artist/gruppe fyller mer enn ett skjerm bilde.)

Klargjøring og forutsetninger

- Programmet godtar at duplikate CD’er forekommer, dvs. vi har ingen sjekk/kontroll på om samme artist er registrert flere ganger med samme tittel.
- Funksjonen “void skriv_til_fil()” skal *ikke* lages som en del av denne eksamensoppgaven.
- Det er plass til alle dataene fra fil i datamaskinens primærhukommelse.
- Gjør dine egne forutsetninger dersom du finner oppgaveteksten upresis eller ufullstendig. Gjør i så fall rede for disse forutsetningene først i besvarelsen din.

Lykke til !

(Etterpå kan du i fred og ro lytte til Dire Straits !)

frode@haug.com

Vedlegg: Halvferdig programkode (.tpl-fil)

```
// INCLUDE:
#include <iostream>           // cin, cout
#include <fstream>            // ifstream, ofstream
#include <cstring>             // strlen, strcmp
#include <cctype>              // toupper
#include <iomanip>             // setw

using namespace std;

// CONST:
const int MAX_CDER          = 500; // Max. antall CD'er i "kartoteket".
const int NVNLEN            = 30;  // Max.lengde for en tekststreng.
const int MAX_SANGER        = 20;  // Max. sang(titler) på en CD.

// STRUCT:
struct CD {                  // Data om en CD:
    int utgivelsesaar;       // - Utgivelsesår.
    char navn[NVNLEN];       // - Artistnavn.
    char tittel[NVNLEN];     // - Tittel.
    int ant_sanger;          // - Antall sanger.
    char sangtittel[MAX_SANGER][NVNLEN]; // - Tittel for sang nr.i.
};

// DEKLARASJON AV FUNKSJONER:
void skriv_meny();           char les();
void les(char t[], char s[]); int les(char t[], int min, int max);
void ny();                   /* Oppgave 3a */ void les_fra_fil(); // Oppgave 3d
void oversikt();             /* Oppgave 3b */ void skriv(); // Oppgave 3e
void fjern();                /* Oppgave 3c */

// GLOBALE VARIABLE:
CD cder[MAX_CDER+1];        // Array av CD-struct'er.
int ant_cder;                // Aktuelt antall CD'er i kartoteket.

int main() {                 // MAIN:
    char valg;
    les_fra_fil();           // Oppgave 3d
    skriv_meny();
    valg = les();
    while (valg != 'A') {
        switch(valg) {
            case 'N': ny(); break; // Oppgave 3a
            case 'O': oversikt(); break; // Oppgave 3b
            case 'F': fjern(); break; // Oppgave 3c
            case 'S': skriv(); break; // Oppgave 3e
            default : skriv_meny(); break;
        }
        valg = les();
    }
    return 0;
}
```

```

// DEFINISJON AV FUNKSJONER:
void skriv_meny() { // Skriver brukerens lovlige valg:
    cout << "\nFølgende kommandoer er lovlige:";
    cout << "\n\tN - Ny CD";          cout << "\n\tO - Oversikt over ALLE CD'ene";
    cout << "\n\tF - Fjern en CD";    cout << "\n\tS - Skriv data om en gitt artist";
    cout << "\n\tA - Avslutt";
}

char les() { // Leser, upcaser og returnerer ETT tegn:
    char ch;
    cout << "\nKommando: "; cin >> ch; // Leser.
    cin.ignore(); // Forkaster neste tegn (='\n').
    return (toupper(ch)); // Upcaser og returnerer.
}

void les(char t[], char s[]) { // Leser inn en tekst med lengde ulik 0.
    do {
        cout << "\t' << t << ": "; // Skriver ledetekst.
        cin.getline(s, NVNLEN); // Leser inn tekst.
    } while (strlen(s) == 0); // Sjekker at tekstlengden er ulik 0.
}

int les(char t[], int min, int max) { // Leser et tall i et visst intervall.
    int n;
    do { // Skriver ledetekst:
        cout << "\t' << t << '(' << min << '-' << max << ")": "; cin >> n; // Leser inn ett tall.
    } while(n < min || n > max); // Sjekker t i lovlig intervall.
    return n; // Returnerer innlest tall.
}

void ny() { // Leser inn alle data om en ny CD:

    // Oppgave 3a: Lag innmaten
}

void oversikt() { // Skriver oversikt over HELE kartoteket:

    // Oppgave 3b: Lag innmaten
}

void fjern() { // Fjerner alle dataene om en gitt CD:

    // Oppgave 3c: Lag innmaten
}

void les_fra_fil() { // Leser inn dataene om CD'ene fra fil og inn i datastrukturen:

    // Oppgave 3d: Lag innmaten
}

void skriv() { // Skriver ALLE data om EN artist, MED eller UTEN sangtitlene:

    // Oppgave 3e: Lag innmaten
}

```