

Laboration 3 – HI1024, Programmering, grundkurs, 8.0 hp

*Dataingenjörsprogrammet, elektroingenjörsprogrammet och medicinsk teknik
KTH – Skolan för Teknik och Hälsa*

Redovisning: Se Kurs-PM om hur redovisningen ska gå till. Läs och följ dessa instruktioner noga!

Den här laborationen är den viktigaste examinationen i denna kurs. Klarar man av att utföra denna självständigt har man de kunskaper som behövs för de fortsatta studierna (kom ihåg att inte hjälpa varandra med koden). Den är krävande och kommer att ta er mycket tid och möda men när ni är klara är ni programmerare, om än ej fullärda. Till er hjälp har ni sist i detta dokument en ledning. Det är nämligen vår erfarenhet att det just nu för de flesta är lite för tidigt att helt själv strukturera upp arbetet runt en sådan här större uppgift. Att använda den är frivilligt. Det är inte tillräckligt att få ihop ett program som utför uppgifterna utan det måste vara bra programmerat. Detta betyder först och främst att det bra uppdelat med hjälp av funktioner och att det använder arrayer och structar där så är lämpligt. Inga globala variabler får förekomma. Vidare ska ni välja bra variabel- och funktionsnamn. Kommentarer i koden bör det finnas en del men ni behöver inte överjobba denna del. Kommentera ganska lite och fokusera istället på att koden i sig blir lättläst.

Uppgiften

I denna laboration skall du skapa ett program för hantering av bankkonton. Varje konto skall representeras av en struct och innehålla kontonummer, ägare och saldo. När programmet startar får användaren ange ett filnamn (med kontouppgifter) och därefter via en meny möjlighet att utföra följande handlingar så länge användaren vill:

- Registrera nya konton
- Skriva ut alla konton
- Sortera konton
- Söka efter konto
- Ändra saldot på ett konto
- Avregistrera ett konto
- Statistik
- Överföring
- Avsluta

Uppstart och avslut

När programmet startar får användaren ange en fil. Finns filen skall programmet läsa in data från filen och spara kontouppgifter i en array av structar (glöm inte att kontrollera att de får plats). När programmet avslutar ska det skriva ut alla kontouppgifter i arrayen till filen. Filen skall vara en textfil (absolut ej binärfil) och får förutom konto-data också innehålla information om antalet konton. Om fil saknas skall programmet börja med en tom array. När programmet avslutas ska det skapa en textfil med det tidigare angivna namnet och där spara ner konto-uppgifterna i arrayen så att de kan läsas in nästa gång programmet körs. Observera att all filhantering sker vid programstart och vid avslut. Däremellan hanterar programmet endast konton i den interna arrayen.

Registrera nya konton

Här har man möjlighet att registrera nya konton. För varje konto anger användaren kontonummer (6-siffrigt nummer) och ägare (för- och efternamn). Saldot sätts till 0 kr. Om kontonumret redan finns får användaren ett felmeddelande och inget konto registreras. Efter att användaren registrerat ett konto (eller misslyckats) får denne direkt möjlighet att registrera ett nytt konto. Om användaren då trycker enter utan att ange kontonummer avslutas registreringen och programmet presenterar åter huvudmenyn. Det är viktigt att vid varje ny registrering kontrollera att arrayen inte är full. Om den är det skall användaren få ett felmeddelande och sedan återgår programmet till huvudmenyn.

Skriva ut alla konton

När användaren väljer detta alternativ skrivs alla konton ut i den ordning de är lagrade i programmet och sedan kommer huvudmenyn upp igen. En utskrift då tre konton är lagrade kan se ut enligt:

Konto-nr	Ägare	Saldo (kr)
000001	AFSHIN FARDI	1900
000002	AMRITPAL SINGH	276304
000003	PETER MAGNUSSON	5000

Sortera konton

Här får man välja på att sortera konton efter konto-nr, ägare eller saldo. Beroende val sorteras kontona i den array de lagras i och programmet återvänder till huvudmenyn. För att se resultatet av sorteringen får man i huvudmenyn välja att skriva ut alla konton. Observera att när man sorterar efter ägare ska programmet helst sortera efter efternamn.

Söka efter konton

Här ska användaren kunna söka efter konton på konto-nr, ägare eller saldo. Vid sökning på konto-nr anger användaren ett konto-nr och får upp rätt kontouppgifter om det existerar och ett felmeddelande om det inte existerar. Vid sökning på ägare ska användaren få upp en lista på alla konton där ägaren innehåller söksträngen i sitt namn. En sökning på PA skulle då kunna ge följande lista:

Konto-nr	Ägare	Saldo (kr)
100234	PATRIK LEM	1900
000002	AMRITPAL SINGH	276304

Vid sökning på saldo får användaren ange ett minvärde och ett maxvärde och får sedan upp en lista på alla konton med saldo mellan dessa värden.

Du får själv välja hur du hanterar stora och små bokstäver när du söker på ägare och programmet behöver inte kunna hantera å, ä eller ö.

Ändra saldot på ett konto

Här får man först välja om man vill ange ett konto-nr direkt eller söka på ägare. Söker man på ägare ska man få upp en lista på samma sätt som när man gjort en sökning på ägare under menyvalet: Söka

efter konto. Efter att man fått upp listan på alla konton som matchar söksträngen får man skriva in kontonumret på kontot man vill ändra saldot för. När man angett saldot ska man få välja om man ska göra ett uttag eller en insättning och därefter ange summan. Om man försöker ta ut mer än vad som finns på kontot nekas uttaget annars ändras saldot.

Avregistrera ett konto

Här får man först välja om man vill ange ett konto-nr direkt eller först söka på ägare precis som för ändra saldo. När man angett konto-nr så ska kontot tas bort från programmet om och endast om det är tomt på pengar. Är kontot inte tomt får man ett felmeddelande.

Det är viktigt att du inte skapar hål i din array när du tar bort ett konto. Det naturligaste sättet att undvika hål är att du flyttar upp alla konton under det avregistrerade ett steg.

Statistik

Om du redovisar denna laboration senast under ordinarie tentamenstillfälle (p1 för data och elektro) ska du hoppa över denna del om du är godkänd på laboration 2. Observera att om du redovisar vid ett senare tillfälle måste du alltid redovisa hela denna del.

När användaren väljer detta val ska denne få en lista på medianen, medelvärdet och typvärdet för saldon i databasen.

Överföring

Om du redovisar denna laboration senast under ordinarie tentamenstillfälle (p1 för data och elektro) ska du hoppa över denna del om du är godkänd på laboration 1. Observera att om du redovisar vid ett senare tillfälle måste du alltid redovisa hela denna del.

Här får användaren möjlighet att göra en överföring mellan två konton. Du behöver inte inkludera någon sökfunktion. Användaren får ange två konton och vilken summa som skall överföras.

Redovisning

Vid redovisningstillfället skall du ha en fil med minst 20 konton redo att läsa in. Dessa bör vara någorlunda realistiska så att man kan pröva de olika funktionerna på ett bra sätt.

Ledning

Först observerar vi att redan instruktionen ger oss följande programflöde:

1. Programmet börjar med att läsa in data från fil till en array av structar (om filen finns).

Programmet stänger sedan filen.

2. Programmet låter användaren välja från huvudmenyn tills denna väljer avsluta. Alla ändringar görs till arrayen. Ingen filhantering utförs.

3. När användaren väljer avsluta skrivs arrayen ut till fil och programmet avslutar.

Nu ska vi dela upp programmet och börja programmera de olika delarna. Här kan man göra många olika val. Nedan följer ett förslag som gör det någorlunda enkelt att testa programmet vartefter. Börjar ni med labben innan ni lärt er någon del kan ni välja en annan ordning men får då lägga lite mera tid på att kunna testa koden.

Arbetsordning

1. Skriv en meny-funktion som du anropar från huvudprogrammet. Denna anropar i sin tur funktioner för de olika alternativen. Börja med att kontrollera att det hela kompilerar och kör utan funktionsanrop. Sedan lägger du till funktionsanrop till funktioner som bara skriver ut en rad med sitt namn. Se också till att man efter ett funktionsanrop får möjlighet att göra ett nytt menyval men att programmet avslutar när man väljer avsluta. Sådär nu har du ett fungerande program som vi kan bygga på bit för bit.
2. Definiera din konto-struct och deklarera en array av sådana structar i main. Sätt storleken på arrayen till 10 000 med hjälp av #define. Deklarera och initiera i main också en variabel som ska hålla reda på antalet konton vi har i vår databas (nedan kallar jag denna nrOfAccounts). Skicka nu med arrayen och en pekare till nrOfAccounts till meny-funktionen och se till att den kan ta emot dessa. Vi skickar här en pekare till antal efter som vi behöver kunna ändra variabelns värde. Se nu till att menyfunktionen kan ta emot parametrarna och kompilera sedan och testkör.
3. Skriv nu funktionen för att registrera nya konton. Även denna behöver arrayen och en pekare till nrOfAccounts (håll tungan rätt i mun med pekarna nu och läs på om du behöver). Använd tillfälliga printsatser för att se att den verkar fungera. Skriv sedan funktionen för utskrift som till att börja med loopar igenom arrayen och väldigt enkelt skriver ut informationen i structarna. Denna behöver arrayen och variabeln nrOfAccounts. Du kan nu köra programmet och lägga till konton och sedan välja skriv ut och se att det fungerar. Förfina nu utskrifterna och rätta eventuella buggar.
4. Vi har nu ett fungerande program som gör något och en funktion som ger oss möjlighet att se vad som händer. Vi ska här nu välja att börja jobba med filhanteringen. Anledningen är att det blir jobbigt att skriva in massa konton varje gång man vill testa sina andra funktioner. Är du inte riktigt redo för filhantering än kan du hoppa detta just nu men får då skriva in konton vid varje testkörning. Vi börjar med funktionen för att avsluta. Skriv denna så att den skriver ut all information i alla structar i vår array till en textfil. Låt den skriva antalet structar överst i filen så blir det lättare att läsa in senare. Namnet på filen kan du just nu hårdkoda. Öppna

filen i en texteditor och kontrollera att det verkar fungera.

5. Det är nu dags att skriva initiera funktionen som ska anropas en gång i början av main.

Det vore här naturligt att läsa in antalet konton i filen och sedan sätta arrayens storlek t.ex. till det dubbla. Pga att vi inte ännu gått igenom allokering blir detta krångligt och svårt att göra i funktioner. Av den anledningen är det mitt råd att du behåller en fast storlek på arrayen (i nuläget 1000 konton) även om detta känns lite primitivt. Vi kan då få en bra uppdelning av programmet i funktioner (om du bemästrar pekare till arrayer och allokering får du gärna allokera minne beroende på antal konton i filen).

Denna funktion skall enligt instruktionen fråga efter filnamn och om det finns en sådan fil lagra alla matcher i filen i vår array. Om filen saknas ska funktionen ta reda på filnamnet som sedan skall användas i avsluta. Ändra nu också funktionen avsluta så att denna skriver till korrekt fil.

6. Nu har vi kommit så långt så att jag tror ni är redo att själva välja i vilken ordning ni jobbar med övriga alternativ. Troligen är funktionen för att ändra saldo en bra startpunkt då den är ganska enkel. Kom ihåg att kompilera och testköra ofta. Ett sista tips: Kolla in funktionen strstr när ni ska jobba med sök-funktionen.

Lycka till!

Nicklas