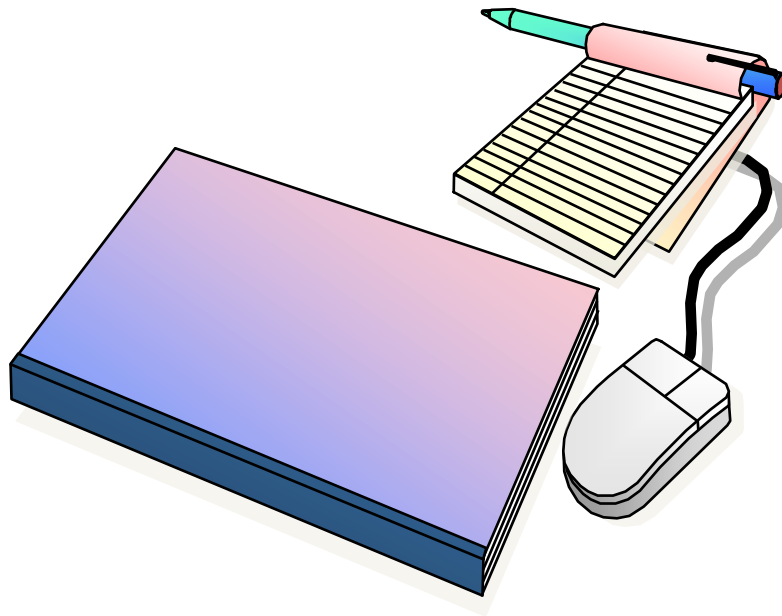


Frequent Flyer



Innehållsförteckning

Frequent Flyer	3
Uppgift 1 – Frequent Flyer File.....	3
Uppgift 2 – Frequent Flyer Program	4
Uppgift 3 – Bonus Report	5
Uppgift 4 – Frequent Flyer Mileage.....	6
Uppgift 5 – Mileage File	7
Uppgift 6 – Mileage Table	8
Uppgift 7 – Date-and-Day Program	9
Uppgift 8 – Airline ID.....	10
Uppgift 9 – Date Search Summary Report.....	12
Uppgift 10 – Destinations Report.....	12
Uppgift 11 – Day-of-Week-Name.....	13
Uppgift 12 – File Update Program.....	13
Uppgift 13 – Mileage Table, sort, pekare.....	14

Frequent Flyer

I den här övningen kommer du att programmera ett antal COBOL-program för att skapa olika rapporter kring flera filer som innehåller information om "Frequent Flyer". Du kommer även att skapa några filer med JCL samt fylla dessa filer med olika data.

- Du kommer att få information om var dina grunddata finns, samt successivt i olika Uppgifter beskrivet vilka program som du skall programmera. Du kommer även att skapa några olika *dataset* samt skriva den JCL som behövs för att förverkliga detta i en IBM COBOL Batch-miljö.

Uppgift 1 – Frequent Flyer File

Denna fil är grunden för flertalet uppgifter i övningen. Filen innehåller information om olika flygresor som en person gjort. Uppgifter som datum (Flight-Date), avrese- och destinationsort (City-Pair), flygbolag (Airline-Id), etc. finns lagrade här.

- Filen har ett utseende som du ser i nedanstående bild.

Kol	Fält	Format
01	Reserved	
02-06	Flight-Date	mm/dd
07-13	City-Pair	xxx-xxx
14-15	Airline-ID	xx
16-19	Flight-Number	nnnn
20	Class-of-Travel	x
21-25	Mileage	nnnnn
26-80	Filler	

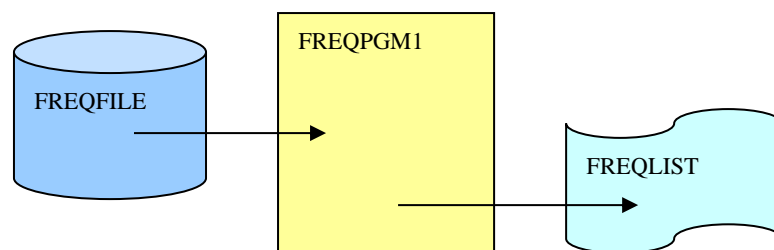
- Innehållet till filen med flygresor finner du på nätet, www.cobolskolan.se/CA01, under rubriken *Frequent Flyer*. Namnet är *Freqfile1*.
- Placera innehållet i denna fil i medlemmen `FREQFIL1` i biblioteket `userid.CA01.COBDATA`.
- Du skall skapa en indexerad fil (VSAM KSDS), där nyckeln skall bestå av `Flight-Date`. Använd programmet `IDCAMS` med funktionen `REPRO` för att fylla filen.
- Filens namn skall vara `userid.CA01.FREQFILE`
- Försäkra dig om att data är sorterat i stigande datumordning genom att exekvera sorteringsprogrammet (Batch) innan du fyller den indexerade filen.

- Gör ett jobb som i Steg1 sorterar filen till ett temporärt dataset.
- I steg 2 exekverar du IDCAMS för att skapa den indexerade filen samt använda den temporära, nu sorterade filen, som input till REPRO.

Uppgift 2 – Frequent Flyer Program

I denna uppgift kommer du att skriva ett program som skapar en rapport från information i resefilen.

- Skriv ett program, FREQPGM1, som läser samtliga poster i filen `userid.CA01.FREQFILE` och skriver en rapport.



- Rapporten skall ha följande utseende:

```

Frequent Flyer Report

Date   Id Fl#   Cl C-Pair   Mileage
-----
01/07  01 0222   Y  DFW-ATL   731
01/10  02 0322   Y  LAS-DFW   1 056
01/11  07 0351   Y  BNA-DFW   631
. . . .
. . .
. .
.
02/23  04 0633   C  SXM-DFW   2 339
02/24  05 0730   Y  RDU-DCA   227
03/10  07 0049   Y  DFW-LAX   1 235
03/16  12 0495   C  SJU-SXM   192
. . .
. .
.

Total Mileage                               nn nnn

Page 1
  
```

- Kompilera och korrigerar tills du har en felfri kompilering.

- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 3 – Bonus Report

- Komplettera ditt program, eller skapa ett nytt med namnet `FREQPGM2` så att rapporten kommer att ha följande utseende:

```
Frequent Flyer Report

Date   Id Fl#   Cl C-Pair   Mileage  Inc Bonus
-----
01/07  01 0222   Y  DFW-ATL   731      731
02/11  03 1609   Y  ATL-BNA   214      500
02/21  07 0351   Y  BNA-DFW   631      631
. . . .
. . .
. .
.
03/03  02 0769   C  SXM-DFW   227      500
04/03  01 1263   Y  RDU-DFW   1 061    1 061
04/10  07 0049   Y  DFW-LAX   1 235    1 235
05/15  11 0428   C  LAX-DFW   1 235    1 544
. . .
. .
.

Total Mileage                mm mmm    nn nnn

Page 1

End of Report
```

- Som Du ser så har det tillkommit en kolumn, *Inc Bonus*. För varje resa så skall en extra bonus tillkomma enligt följande regler:
 - Samtliga klasser skall alltid få minst 500 miles när reslängden är lägre än 500, men ingen ytterligare bonus skall då tillkomma.
 - Turistklass, 'Y', skall inte ha någon extra bonus.
 - Business Class, 'C' skall erhålla 125% på aktuell reslängd.
 - First Class, 'F' skall erhålla 150% på aktuell reslängd.
- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 4 – Frequent Flyer Mileage

I denna uppgift så kommer du att skapa en ny indexerad fil som skall fyllas med data, så att du från denna kan hämta reseavstånd (Mileage) när du senare skall skapa din rapport. Filen FREQFILE förväntas inte innehålla Mileage.

- Skapa en VSAM KSDS-fil som skall fyllas med data från den fil som Du hittar på www.Cobolskolan.se/CA01 och den filen *FreqMile*.
- Primärnyckel skall vara *City-Pair*.
- Hantera denna fil på samma sätt som den filen i Uppgift 1, d.v.s. den skall vara sorterad på nyckelbegreppet, som är *City-Pair*, innan du fyller den nya filen.
- Namnet på den filen skall vara *userid.CA01.FREQMILE*.
- Filen har ett utseende som liknar denna:

Kol	Fält	Format
01-07	City-Pair	xxx-xxx
08	Filler	x
09-13	Mileage	nnnnn
14-80	Filler	x

- Innehållet ser du delar av här:

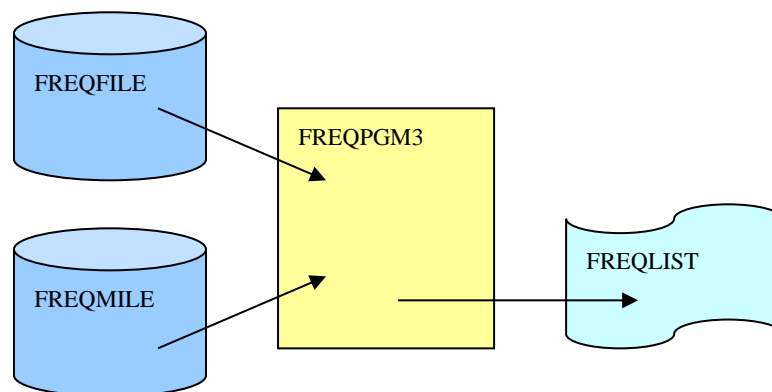
```
AKL-HNL/04406
AKL-MEL/01636
AKL-NAN/01341
AKL-PPT/02544
AKL-SYD/01343
. . .
DFW-DEN/00645
DFW-DSM/00624
DFW-DTW/00987
DFW-ELP/00553
. . .
SYD-NLK/01043
YYZ-DFW/01199
YYZ-LGA/00356
YYZ-ORD/00437
```

- Notera att filen bara innehåller information om flygväg i ena riktningen, t.ex. DFW-DSM, men `FREQFILE`-filen kan mycket väl innehålla information om en resa gjord i andra riktningen.
- Kontrollera resultatet i File Manager. Du kan skapa en COBOL *Copy-Book* som beskriver hur posterna ser ut, så ser Du bättre att allt är OK.

Uppgiften är avslutad.

Uppgift 5 – Mileage File

- Skapa ett nytt program med namnet `FREQPGM3`, med `FREQPGM2` som mall.
- Ändra programmet så att det hämtar reseavståndet från den nyligen skapade filen, `userid.CA01.FREQMILE`, och inte från posterna i filen `userid.CA01.FREQFILE`.



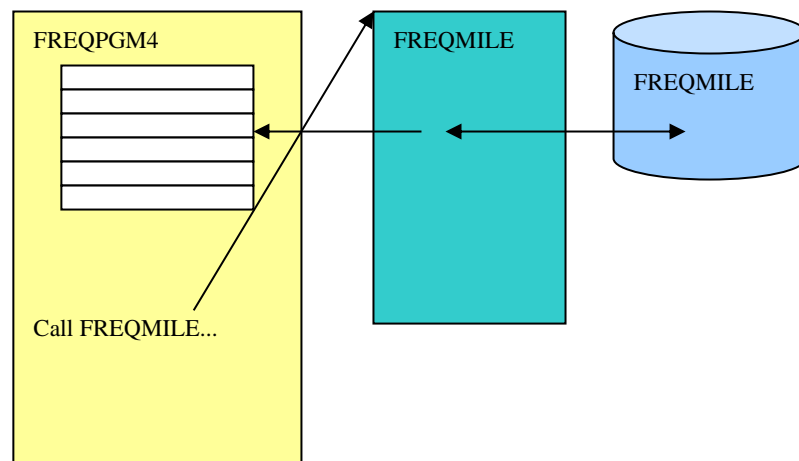
- Kompilera och korrigeras tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 6 – Mileage Table

I denna uppgift så kommer du att skriva ett program som läser in filen med reseavstånden till en intern COBOL-tabell, så att du får söka i denna tabell för att få fram avstånd.

- Skriv ett nytt program, FREQMILE, som läser in filen `userid.CA01.FREQMILE` till en intern COBOL-tabell. Tabellen skall finnas fysiskt i huvudprogrammet, som du tidigare skrivit. Räkna med max 400 tabellelement. Du skall anropa "tabellprogrammet" från huvudprogrammet för att fylla tabellen med data.
- Kompilera och korrigerar tills du har en felfri kompilering.
- Kopiera det tidigare programmet FREQPM3 till ett nytt program med namnet FREQPM4. Korrigera sedan huvudprogrammet, så att det anropar programmet FREQMILE för att fylla tabellen.
- Programmet FREQPM4 skall söka i tabellen för att finna reseavstånd.



- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

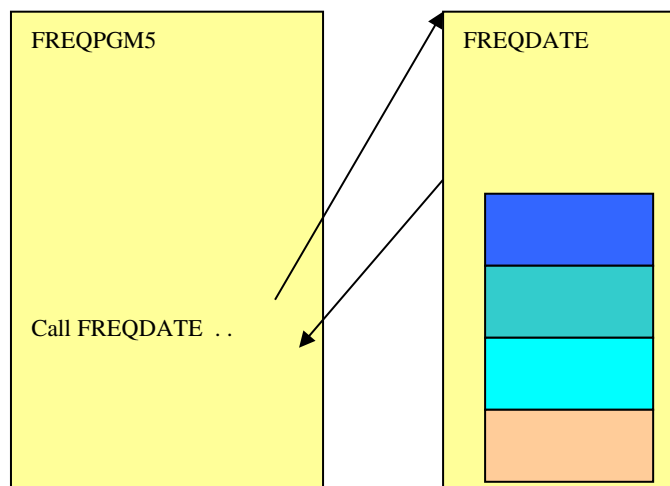
Uppgift 7 – Date-and-Day Program

I den här uppgiften skall du skriva ett program som skall kunna skapa datum i olika format. Programmet skall kunna anropas med en parameter som beskriver i vilket format datumet skall levereras.

- Skriv ett program, FREQDATE, som skall kunna anropas som ett subprogram, där du i anropet skall kunna ange att det returnerade datumet skall vara i något av nedanstående format.

```
DATE=S    ( 6/4, 20xx )  
DATE=M    ( 6 April, 20xx )  
DATE=L    ( Måndag 6 April, 20xx, kl hh:mm )
```

- Programmet skall konstrueras så att det består av ett "huvudprogram" som analyserar att anropsparametern är korrekt. Sedan skall de olika datumformaten konstrueras av "inbakade" program i samma laddmodul/källkod.



- Lämpligen skapar du ett huvudprogram som innehåller enskilda program för att skapa de olika datumformaten.
- Bestäm själv formatet för interna anropsparametrar etc. Datum som programmet skall returnera är dagens datum samt ett framtida datum om 30 dagar.
- Komplettera sedan huvudprogrammet så att det kan anropas med en parameter vid exekveringen (DATE=S / DATE=M / DATE=L). Programmet skall även skriva ut ett meddelande om exekveringsparameter utelämnats eller är felaktig och avslutas med lämplig returkod.
- Uppdatera programmet, så att rapporten får ett utseende som liknar denna:

```
Frequent Flyer Report

Printed      :      Måndag, 6 April, 20xx
Valid Until  :      Onsdag, 6 Maj, 20xx

Date   Id   Fl#  Cl C-Pair  Mileage  Inc Bonus
-----
01/07  01   222  Y  DFW-ATL    731      731
01/10  02   322  Y  LAS-DFW   1056     1056
01/11  07   351  Y  BNA-DFW    631      631
01/15  11   428  C  LAX-DFW   1235     1543
01/16  05  1459  F  TPA-SJU   1237     1855
. . . .
. . .
. .
.
03/24  06  1088  Y  DFW-RDU   1061     1061

Total Mileage                13430     17129
```

- Kompilera och korrigerar tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 8 – Airline ID

- Komplettera programmet `FREQPGM6` så att det i rapporten skriver ut namnet på flygbolaget. Innehållet till filen med namnen på flygbolagen finner du på nätet, www.cobolskolan.se/CA01, under rubriken *Frequent Flyer*. Namnet är *Freqline*.
- Placera innehållet i denna fil i medlemmen `FREQLINE` i biblioteket `userid.CA01.COBDATA`.
- Du skall skapa en indexerad fil (VSAM KSDS), där nyckeln skall bestå av `Airline-Id`. Använd programmet `IDCAMS` med funktionen `REPRO` för att fylla filen.
- Filens namn skall vara `userid.CA01.FREQLINE`
- Försäkra dig om att data är sorterat i stigande datumordning genom att exekvera sorteringsprogrammet (Batch) innan du fyller den indexerade filen.
- Filen har följande utseende:

Kol	Fält	Format
01-02	Airline-Id	nn
03	Filler	x
04-29	Airline-Name	xxxx...xxx (25)
30-80	Filler	x

- Det kan vara lämpligt att läsa in denna fil i en intern COBOL-tabell och söka i denna för att hämta *Airline-Name*.
- Räkna med att maxantalet flygbolag är 25.

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 9 – Date Search Summary Report

- Skriv ett program, `FREQPGM7` som kan skriva ut en rapport över vilka resor som gjorts under en viss tidsperiod.
- Läs in fr.o.m. datum – t.o.m. datum från en datumfil. Bestäm själv namn på filen, men enklast är att ange sökdatum som *instream-data* i din exekverings-JCL.
- Formatet skall vara mmdd-mmdd. Denna datumgräns kan Du ange som *Instream-Data* i din JCL vid exekveringen.
- Programmet skall kontrollera att formatet är korrekt.
- Om formatet är felaktigt, så skall programmet avsluta med returkod 99.
- Om formatet är korrekt, men sökvärdena är felaktiga, så skall programmet avslutas med returkod 98.
- Skulle filen saknas, eller vara tom, så skall detta accepteras, och då skall samtliga poster skall skrivas ut i rapporten.
- Rapporten skall visa vilka flygturer som gjorts under en viss period med tidpunkt enligt angivna sökdatum.
- Rapportens utseende skall ha likheter med den tidigare rapporten.

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 10 – Destinations Report

- Skriv ett program, `FREQPDST` som kan skriva ut en rapport över vilka resor som gjorts med ett visst City-Pair. Du kan använda `FREQPGM7` som mall.
- Läs in *City-Pair* via `SYSIN`. Flera *City-Pair* skall kunna anges som olika poster/rader.
- Programmet skall även kunna hantera situationen när endast avrese- eller destinations-ort anges, så att man kan se vilka resor som gjorts från eller till en viss stad.

- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

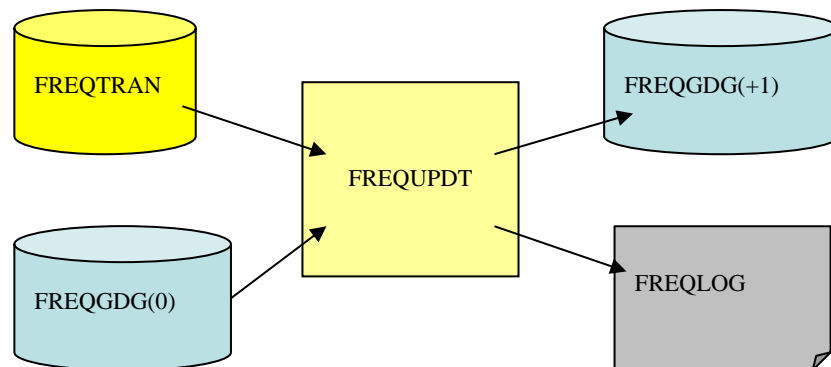
Uppgift 11 – Day-of-Week-Name

- Komplettera rapporten så att den innehåller dagnamn när resan gjorts.
- Du kan använda programmet `FREQPGM6` som mall. Ge programmet namnet `FREQPG11`.
- Du kommer att behöva komplettera datum-programmet, `FREQDATE`, så att det kan leverera ett dagnamn med ledning av datum. Antag att året är innevarande år. Ge programmet namnet `FREQDATX`.
- Kompilera och korrigeras tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.

Uppgiften är avslutad.

Uppgift 12 – File Update Program

I denna uppgift så kommer du att skriva ett program som läser in en transaktionsfil med information om förändringar som skall göras i resefilen. Innan programmet gör uppdateringar, så skall Du med JCL skapa en backupkopia av resefilen som en ny generation i ett GDG.



- Skapa en generationsdatasetgrupp med namnet `userid.CB01.FREQGDG`.
- Gruppen skall innehålla maximalt 10 generationer, och föråldrade generationer skall raderas automatiskt.
- Gör samtidigt en `LISTCAT` för att kontrollera att gruppen har de egenskaper som du tänkt dig.
- Skriv sedan ett program, `FREQUPDT`, som läser in filen `FREQTRAN`, `userid.CA01.FREQTRAN`. Denna skall innehålla poster med transaktioner som skall appliceras på den aktuella resefilen.

- Data till transaktionsfilen finner Du på nätet, www.cobolskolan.se/CA01, under rubriken *Frequent Flyer*. Namnet är *Freqtran*.
- Transaktionerna har följande utseende:

```
Pos 1      : A (Add)
Pos 2-80   : Komplet format, som du ser i Uppgift 1.

Pos 1      : R (Remove)
Pos 2-6    : Flight Date

Pos 1      : U (Update)
Pos 2-6    : Flight Date
Pos 7      : Flight Class
```

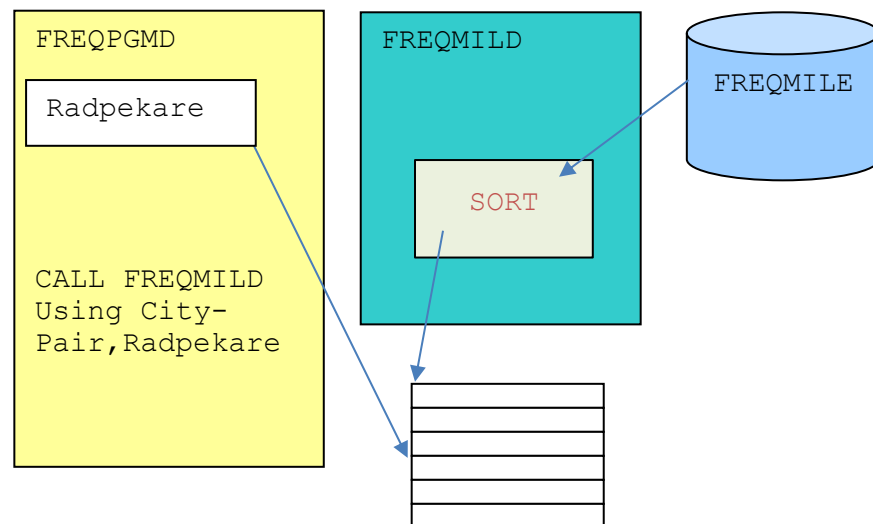
- Programmet skall kontrollera att datumet i transaktionerna är korrekt. (se:LE Call CEEDAYS)
- Programmet skall även skriva en loggfil, *FREQLOG*, med information om gjorda transaktioner.
- Felaktiga transaktioner skall även skrivas till loggfilen, så att dessa kan korrigeras/följas upp.
- När felaktiga transaktioner upptäcks, så skall programmet avslutas med returkod 100. I detta fall så skall resefilen återställas från den sparade backup-kopian i GDG-datasettet.
- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.
- Kontrollera resultatet och jämför med en tidigare rapport samt transaktionerna i *FREQTRAN*.
- Uppgiften är avslutad.

Uppgift 13 – Mileage Table, sort, pekare

I denna uppgift så kommer du att använda pekare och LE-call för att förändra programmen i Uppgift 6.

- Programmet *FREQMILD* skall använda filen med reseavstånd för att skapa en tabell. Använd programmet *FREQMILE* som mall.
- Innehållet till filen med reseavstånden finner du på nätet, www.cobolskolan.se/CA01, under rubriken *Frequent Flyer*. Namnet är *Freqmile*.

- Placera innehållet i denna fil i medlemmen FREQMILE i biblioteket *userid.CA01.COBDATA*.
- Filen antas vara osorterad och programmet skall göra en COBOL-internsort av filen innan tabellen fylls med data. Nyckeln är som tidigare *City-Pair*.
- Utrymme för tabellen skall allokeras genom att utnyttja *LE Storage Management-call*, *CEEGTST*, eller med COBOL **Allocate**. Storleken på tabellen avgörs av innehållet i filen FREQMILE.
- Programmet FREQPGMD skall anropa FREQMILD med begäran om sökning av visst *City-Pair* samt skicka med en pekare (*pointer*), som detta program skall uppdatera så att den pekar på den sökta posten i tabellen vid lyckad sökning. Använd programmet FREQPGM6 som mall till programmet FREQPGMD.
- Om sökningen misslyckas skall pekaren ha värdet Null.



- Kompilera och korrigera tills du har en felfri kompilering.
- Exekvera programmet och kontrollera resultatet.
- Resultatet skall ju ej bli annorlunda jämfört med uppgift 6, men programmen innehåller några annorlunda tekniker.
- Uppgiften är avslutad.

Övningen är avslutad.