2016-01-15.kl.08-13

Tentaupplägg

TIPS 1:

Läs igenom ALLA uppgifterna. Välj den du känner är lättast först. Det kan gärna ta 10-20 minuter. Försök skriva saker som kan vara problem i uppgifterna. Är det något du absolut kommer att fastna på så kanske det är fel uppgift att ge sig på. Tiden du lägger på att noga läsa uppgifterna tjänar du in på att välja rätt uppgift.

TIPS 2:

Kolla ibland till kommunikationsfönstret. Det kan ha kommit information till alla utan att ni skickat in en fråga. Kanske gäller det dig också (d.v.s. den uppgift du jobbar med).

TIPS 3:

Om ni har problem med kompilator, Emacs eller annat som INTE har med uppgifterna att göra, räck upp handen så kommer en assistent. Detsamma gäller om hur man kopierar givna filer
" cp given_files/* . " eller liknande.

Frågor om själva uppgifterna tar vi i första hand via tentasystemet.

I körexemplen har vi markerat det som användaren matar in på tangentbordet med ***fet och kursiverad*** stil. Tänk på att körexemplen bara är *ett* exempel på när programmet körs. Testa ditt program noga och tänka över hur programmet skall fungera vid andra indata.

Vi hinner normalt sett inte svara på frågor de sista 10 utminutera på tentan. Då ägnar vi all tid åt att rätta uppgifter för att alla skall hinna få svar innan ni går hem.

Betygsgränser:	**Tid**	**DI/EL**	**SVP**
1 poäng	13:00	Betyg 3	Betyg G
3 poäng	12:00	Betyg 4	
3 poäng	10:45		Betyg VG
3 poäng	10:30	Betyg 5	
4-6 poäng	12:00	Betyg 5	
4-6 poäng	12:15		Betyg VG

Bonustid från laborationerna tillkommer till dessa tidsgränser. Tiden för betyg 3/G överstiger dock aldrig 5 timmar.

OBS: Delpoäng delas inte ut på uppgifterna. För att få poäng på en uppgift måste man alltså lösa uppgiften helt (och enligt specifikation).

Lycka till med tenterandet och hoppas att alla får G på minst en uppgift idag.
M.v.h.
/Torbjörn (examinator)

Uppgift 1 - IMDb [1p]

En student Kim på kursen TDIU08 har blivit sugen på att göra sin egen variant på jättesajten IMDb, kallad *Movie Rater*. Kim har tänkt att Movie Rater skall ha många funktioner som IMDb inte har, och på detta sätt skall Movie Rater bli mer populärt än IMDb. I nuläget har Kim dock bara hamrat ut ett huvudprogram. Det skall gå att lägga till filmer i Movie Raters "databas", och man skall kunna skriva ut en lista på alla filmer som finns i databasen. En häftig funktion är också att man kan avsluta programmet! Du kan se hur långt Kim har kommit i filen movie_rater.cpp, som ligger i mappen given_files.

Det saknas dock lite kod i Kims program för att det skall fungera som Kim har tänkt. Huvudprogrammet är färdigt, du får inte ändra på det. Kim använder en vector för själva "databasen" men du behöver lägga till datatypsdefinitionen för *Movie_Type*, och följande funktioner:

- *get*, som läser in data från standardinmatningsström till en variabel av typen Movie_Type. Användaren matar in filmens produktionsår, betyg och titel. Se körexemplet för hur Kim har tänkt sig att filmerna matas in.
- *add*, som lägger till en Movie_Type till "databasen".
- *print_all*, som skriver ut alla filmer i databasen till standardutmatningsström. Se körexemplet för hur Kim har tänkt sig att filmerna skall skrivas ut. Betyget skall alltid skrivas ut med en decimal, oavsett hur användaren matade in.

Körexempel:

```
Welcome to the Movie Rater

Please make your selection:
1. Enter new movie
2. Print movies
3. Quit
>> 1
Please enter year, rating and name for the movie:
1999 8.9 Fight Club

Please make your selection:
1. Enter new movie
2. Print movies
3. Quit
>> 1
Please enter year, rating and name for the movie:
1994 6.748 Forest Gump

Please make your selection:
1. Enter new movie
2. Print movies
3. Quit
>> 1
Please enter year, rating and name for the movie:
2015 10 Star Wars - the Force Awakens

Please make your selection:
1. Enter new movie
2. Print movies
3. Quit
>> 2
Fight Club (1999), rating: 8.9
Forest Gump (1994), rating: 6.7
Star Wars - the Force Awakens (2015), rating: 10.0

Please make your selection:
1. Enter new movie
2. Print movies
3. Quit
>> 3
```

TIPS: Du får dela upp ditt program så att du använder en h-fil, men detta är inte ett krav. Om du delar upp i fler filer så måste du skicka alla filer vid rättning.

2016-01-15.kl.08-13

Uppgift 2 - Registreringsnummer [1p]

Regskylt med skattemärke. Skattemärket används inte längre dock...

Skriv ett program som läser igenom filen REGNUMMER.TXT och kontrollerar att varje registreringsnummer i filen är:

- är sex tecken långt.
- inleds med tre alfabetiska tecken.
- avslutas med tre numeriska tecken ('0' - '9').

Om det är fler än ett fel på regnumret så skriver man bara ut det första man stöter på. Man gör kontrollen i ordningen som kraven står ovan.

De registreringsnummer som inte är korrekta skall skrivas ut av programmet, följt av en text som beskriver vad som är fel. De registreringsnummer som är korrekt följs bara av texten "OK!"

Det finns fler kontroller som man skulle kunna göra. Man får t.ex. inte ha fula/stötande ord på regskylten, som t.ex. "MES", "OND" och "SPY". Vi gör dock inga sådana kontroller i denna uppgift.

Körexempel:

```
WNF766 OK!
DP309 Regnumret är för kort!
ARKF191 Regnumret är för långt!
SOM123 OK!
19KF32 Regnumret inleds inte med alfabetiska tecken!
KPR9B1 Regnumret avslutas inte med numeriska tecken!
LAT666 OK!
```

KRAV: Lös problemet med kontrollerna generellt. Om man skulle vilja modifiera programmet så att det klarar t.ex. fyra bokstäver följt av fyra siffror så skall man bara behöva ändra på enstaka ställen i din kod.

TIPS: Dela upp kontrollerna i underprogram.

Uppgift 3 - Hockey-VM [2p]

På filen MATCHES.TXT finns det resultat från senaste junior-VM i ishockey. Varje rad i filen motsvarar en match och har följande format:

LAG vs LAG M1-M2 S1-S2

Där LAG är tre tecken som beskriver ett land, t.ex. "SWE" motsvarar Sverige och "RUS" motsvarar Ryssland. M1 och M2 är antal mål som lagen gjorde. S1 och S2 är antalet skott på mål. I denna uppgift skall du skriva ett program som läser igenom filen och för varje lag skriver ut dess målprocent, d.v.s. totalt antal gjorda mål genom totalt antal gjorda skott.

Det är meningen att du i denna uppgift skall lagra det data du behöver för att lösa problemet i en intern datastruktur i ditt program (d.v.s. du skall endast läsa igenom filen en gång). Exakt hur datastrukturen ser ut får du själv bestämma.

Körexempel:

```
TEAM  GOAL-RATIO
----------------
CZE        7.02%
RUS       10.95%
SUI       10.00%
SWE       11.28%
FIN       15.22%
BLR        7.38%
USA       14.29%
CAN       10.11%
SVK        6.50%
DEN        8.75%
```

TIPS: Skilj på heltals- och flyttalsdivision.

Uppgift 4 - Bollar och glasrör [2p]

Vi tänker oss ett antal genomskinliga glasrör stående på rad. Som ett spel kan man tänka sig att vi släpper ner bollar från ovan och låter dem landa i rören. Vi tänker oss att det är slumpmässigt vilket rör som just varje boll hamnar i. Vi antar även att rören är väldigt långa och smala. Skriv ett program där användaren får mata in antal rör (maximalt 99 stycken). Användaren skall därefter mata in hur många bollar vi vill släppa ned (också maximalt 99 stycken). Programmet skall därefter slumpa hur bollarna hamnar i rören och rita ut detta på skärmen. I den utritade "bilden" skall även rören vara numrerade från 1 upp till antal rör. Ingen felhantering av indata krävs.

Körexempel 1:

```
Mata in antal rör: 5
Mata in antal bollar: 13

        o
    o   o
    o   o  o
    o   o  o
  o o   o  o
  1 2 3 4  5
```

Körexempel 2:

```
Mata in antal rör: 4
Mata in antal bollar: 7

      o
      o  o
  o o o  o
  1 2 3  4
```

Körexempel 3:

```
Mata in antal rör: 13
Mata in antal bollar: 7

                        o
  o o             o     o     o     o
  1 2 3 4 5 6 7 8 9 10 11 12 13
```