

# 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 minuterna på tentan. Då ägnar vi all tid åt att rätta uppgifter för att alla skall hinna få svar innan ni går hem.

**OBS:** Tänk på att uppgifterna är numrerade 5-8. Det är viktigt att du skickar in din lösning på rätt uppgiftsnummer.

<b>Betygsgränser:</b>	<b>Tid</b>	<b>TekFak</b>
1 poäng	17:00	Betyg 3
2 poäng	16:30	Betyg 4
2 poäng	15:00	Betyg 5
>=3 poäng	16:00	Betyg 5

Bonustid från laborationer tillkommer till dessa tidsgränser. Tiden för betyg 3/G överstiger dock inte fyra timmar.

Lycka till med tenterandet och hoppas att alla får G på minst en uppgift idag.

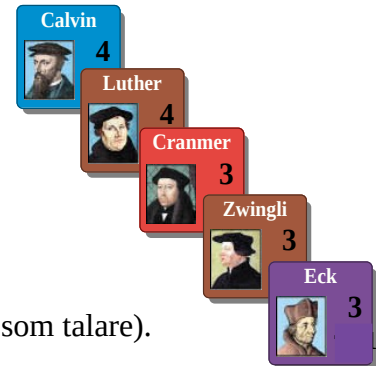
M.v.h.

/Torbjörn (examinator)



## Uppgift 6 – Teologisk Debatt [1p]

På femtonhundratalet skedde det många teologisk debatt efter att Martin Luther dragit igång reformationen. Du skall i denna uppgift simulera en teologisk debatt mellan katolska och protestantiska talare.



På filen DEBATERS.TXT finns det data som representerar teologiska debattörer. Varje sådan har följande delar:

- Ett namn (en sträng).
- En religiös tillhörighet (ett tecken, 'K' för katolik, 'P' för protestant).
- Ett färdighetsvärde (heltal mellan 1 och 3, hur pass duktig personen var som talare).

Du skall skapa ett program som definierar en struct `Debater_Type` som kan lagra datat för **en** debattör. Programmet skall lagra **alla** debattörerna i **en** Vector av `Debater_Type`. Programmet måste inte läsa från filen utan det är okej att hårdkoda datat rakt in i programmet. Programmet skall sedan slumpa fram en katolsk eller en protestantisk debattör och presentera personens data för användaren.

**TIPS:** Det finns ett givet program, `dice.cpp` som slumpar fram ett tärningsslag.

### Körexempel 1:

Protestantisk debattör: Luther (4)

### Körexempel 2:

Katolsk debattör: Eck (3)

**OBS:** Eftersom programmet slumpar så skall det (troligtvis) bli olika utgång varje gång man kör programmet.

## Uppgift 7 – Boka dagar i framtiden [2p]

Skapa ett program som sköter en kalender där man kan boka hela dagar. Man skall alltså kunna boka hela dagar i kalendern fr.o.m. 2020-01-01 t.o.m. år 9999-12-31. Vi antar att användaren alltid matar in korrekta datum. Huvudprogrammet skall låta användaren göra bokningar genom att skriva in datum. Om användaren skulle försöka boka samma dag två gånger skall ett felmeddelande skrivas ut. Programmet avslutas genom att användaren trycker enter och sedan ctrl-d.

**KRAV 1:** Du skall ha en struct `Date_Type` i ditt program som innehåller tre heltal. Använd denna för att lagra datum. Du skall ha en funktion **get** som läser in data till **en** variabel av typen `Date_Type`. Datatypen skall definieras på en separat headerfil `date_handling.h` med en tillhörande implementationsfil `date_handling.cpp`.

**KRAV 2:** I huvudprogrammet skall du använda vector på något sätt.

### Körexempel:

```
Mata in datum: 2021-04-15
Bokning gjord!
Mata in datum: 4040-03-10
Bokning gjord!
Mata in datum: 2021-04-15
Kan ej göra bokning, detta datum är redan bokat!
Mata in datum: 6543-11-28
Bokning gjord!
Mata in datum:
[ctrl-d]
```

## Uppgift 8 – Nya Världen [2p]



Fjorton- och femtonhundratalet var verkligen en tid för upptäcktsresande! Du skall i denna uppgift simulera upptäcktsresor för de relevanta stormakterna Spanien, Frankrike och England.

På filen EXPLORERS.TXT finns det data som representerar upptäcktsresande personer. På varje rad i filen finns det ett efternamn (t.ex. Roberval), en sträng som representerar vilket land personen kommer ifrån och ett heltal (0-4) som anger hur duktig personen var på att upptäcktsresande.

På filen OUTCOMES.TXT finns det lite händelser som kan ske när upptäcktsresarna gör sina resor. En på varje rad.

Du skall i denna uppgift skriva ett program som slumpar ett tal mellan 1 och 12 och tar motsvarande person från EXPLORERS.TXT. Därefter skall programmet slumpa ett tal mellan 1 och 6 lägga till personens heltal. Detta ger ett resultat mellan 1 och 10. Programmet skall sedan ta motsvarande rad från OUTCOMES.TXT och skriva ut detta i terminalen.

### Körexempel 1:

Chancellor åker ut på expedition för England! Tärningsslag  $3 + 1 = 4$   
Ingen ny upptäckt, återvänder hem.

### Körexempel 2:

DeSoto åker ut på expedition för Spanien! Tärningsslag  $5 + 2 = 7$   
Mississippifloden upptäcks.

### Körexempel 3:

Cartier åker ut på expedition för Frankrike! Tärningsslag  $2 + 3 = 5$   
St. Lawrencefloden upptäcks.

### Körexempel 4:

Magellan åker ut på expedition för Spanien! Tärningsslag  $6 + 4 = 10$   
Världsomsegling!

**KRAV:** Du får **inte** lagra hela innehållet i textfilerna i datorns minne. Läs lite i taget.

**OBS:** Eftersom programmet slumpar så skall det (troligtvis) bli olika utgång varje gång man kör programmet. Det betyder även att ditt programs utdata kommer skilja mot ovanstående körexempel.