

Tentamen

Detta dokument informerar om hur tentamen är uppbyggd. Lösningen till tentamen lämnas in individuellt. Det är **inte tillåtet** att arbeta eller diskutera tillsammans med andra personer. Alla studenter som gör detta under tentamen kommer att bli underkända oavsett om ni gav eller fick hjälp.

Det är dock tillåtet att titta på tidigare övningar och anteckningar samt söka information på webben.

Zoom under tentamen

Det är viktigt för oss att kunna nå er alla snabbt under tentamen. Alla som gör tentamen (inte exempeltentamen) måste vara uppkopplade på zoom (**använd zoom-länken som Sebbe har använt under kursen**) under tentamen (10:00 - 12:00). Alla ska ha kameran AV och vara mutade, även Sebbe. Men om Sebbe säger något ska ni alla höra det direkt. Han kommer bara att använda zoom om det är något akut som han upplever att ni alla behöver veta om.

Ingen student får använda zoom för att ställa frågor eller så (så ni inte stör andra). Det gör ni som vanligt via email (sebastian.bengteward@mau.se). Använd inte chatten för vi kommer inte att kolla på den.

När du har lämnat in och är klar med tentan kan du också lämna zoom.

Redovisning

På grund av COVID-restriktionerna består tentamen av två delar: först ska du lämna in svaret på frågorna (**kl. 10-12 på torsdag**) och sedan ska du redovisa dina svar individuellt för Sebbe eller Erik. Redovisningen går till så att vi ses via Zoom (vi lägger upp ett schema och kontaktar er kort efter tentan), läraren delar skärmen, visar din kod och ställer frågor om den. Om du själv har skrivit koden ska det inte vara något som helst problem att svara på frågorna. Om du inte kan svara på frågorna blir du underkänd i tentamen.

Om man inte har G i koden man lämnar in (se nedan) fyller inte redovisningen någon funktion. En lysande redovisning kan inte på egen hand göra så att du blir godkänd i tentamen.

Om du är säker på att du inte har G i tentan går det bra att stryka sig från redovisningen. Vi kommer trots det att rätta din kod och skulle det sedan mot förmodan visa sig att du fick G på tentan går det alltid att senare fixa en redovisning för dig.

Upplägg

Denna tentamen består av **ett dataset** samt frågor kopplade till detta. För varje fråga ska du skriva **en funktion** som tar emot ett antal parametrar och returnerar något. **Alla funktioner** tar emot detta dataset-objekt. Ni hittar dataset-objektet längst ner i dataset-filen (dataset-1.js). Skulle en funktion behöva ta emot fler parametrar står det tydligt beskrivet i frågan. Vad funktionen ska returnera, t.ex. en siffra eller en array, står också tydligt beskrivet i frågan.

Frågorna är uppdelade i A-, B- och C-frågor (med stigande svårighetsgrad). A-frågor ger 1 poäng, B-frågor ger 2 poäng och C-frågor 4 poäng. Namnen på era funktioner ska vara kopplade till vilken fråga dom besvarar, dvs. funktionen till A-frågan 2 ska vara *A2*, funktionen till B-frågan 4 ska vara *B4*, och så vidare.

Betyg

För att få G måste du lämna in korrekta svar på frågor som totalt ger minst 8 poäng (det går såklart bra att lämna in så många lösningar som du vill). Minst en av frågorna som du har svarat korrekt måste vara en B-fråga. Du har runt 15 minuter per fråga (7 frågor ska besvaras). Vi bedömer att det räcker gott för att skriva den förhållandevis korta koden som behöver skrivas, om man känner sig säker på array-metoderna. Det kan dock bli ont om tid om man ska testa sig fram till en lösning som fungerar.

Du behöver endast lämna in en JavaScript-fil med funktionerna som du har skrivit. För att testa dem under tentan behöver du dock skapa en HTML-fil. Den bygger du bäst du vill. Glöm inte att inkludera dataset-filen så den är tillgänglig (annars kan ni inte testa era funktioner). Din JavaScript-fil (endast en) lämnas in på Canvas (se nedan).

A-Frågor

1. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla ID:en på de tre personer med lägst budget.
2. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla för- och efternamn (sammanslaget i en sträng) på de personer som ska handla 3 varor.
3. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla ID:en på de personer som har en budget som är mindre än 20.
4. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en sträng. Strängen ska innehålla namnet på den dyraste varan (om det finns flera varor som har samma pris räcker det med att ni tar den första).
5. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla namnet på de varor som inte har ett korrekt pris (alltså där priset är null).
6. *Dataset 1*: Funktionen ska ta emot *dataset 1* **och** en sträng samt returnera ett objekt. Funktionen ska hämta den vara (hela objektet) vars namn är det samma som sträng-argumentet (t.ex. "Blackberries").
7. *Dataset 1*: Funktionen ska ta emot *dataset 1* **och** en siffra samt returnera ett objekt. Funktionen ska hämta den person (hela objektet) vars ID är det samma som siffer-argumentet.

B-Frågor

1. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla för- och efternamn (sammanslaget) på de personer som ska handla 4 eller fler varor.
2. *Dataset 1*: Funktionen ska ta emot *dataset 1* **och** en siffra samt returnera en siffra. Funktionen ska räkna ut och returnera den totala kostnaden för en persons inköpslista (vilket är den personen vars ID är det samma som siffer-argumentet).

3. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla för- och efternamn (sammanslaget i en sträng) på de personer som ska köpa varan med namnet "Pepper - Jalapeno".
4. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla ID:en på de personer som ska köpa varan med ID:et 37.

C-Frågor

1. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en array. Arrayen ska innehålla för- och efternamn (sammanslaget i en sträng) på de personer vars budget inte räcker till. (De ska alltså köpa varor för mer pengar än de har budget)
2. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en sträng. Strängen ska innehålla för- och efternamn (sammanslaget i en sträng) på den person som har den dyraste inköpslista (om flera personer har samma konstnad på sin inköpslista räcker det med att ni tar den första).
3. *Dataset 1*: Funktionen ska ta emot *dataset 1* och returnera en siffra. Siffran som returneras är medelvärdet (kostnaden) för alla personers inköpslistor (medelvärdet = summan / antal).

Inlämning

Klockan **12:05** kommer uppgiften på Canvas att stänga. Inga inlämningar kommer att accepteras som inte finns på Canvas så planera inlämningen väl. Alltså börja fixa med inlämningen **senast** klockan 11:55.

Skickar du in flera versioner kommer den senaste gälla, så tänk på att få med allting i den senaste versionen (dvs. inkludera allt från föregående version också).

Om Canvas ger dig problem vid inlämningen så går det bra att skicka ett email till sebastian.bengteward@mau.se med din JavaScript-fil, **skriv tydligt vilken kurs och tentamen det gäller**.

Skulle du få problem med att mejla din JavaScript-fil (gmail brukar ge problem) så kan du använda dig av tjänsten wetransfer.com.

Inlämningar utanför Canvas (email eller wetransfer) **måste** vara skickade **före** klockan 12:05 (tiden är avgörande här).

Nästan korrekta svar

Vi har redan diskuterat detta i samband med första webbkursen. Utgångspunkten är att svaren ni lämnar in ska fungera utan problem exakt som beskrivet i frågan. Lämnar ni kod som inte fungerar perfekt så kommer vi att göra en bedömning om huruvida ni kan få några poäng eller inte. Den bedömningen beror inte bara på den specifika frågan utan även på andra svar i tentamen. Detta är en av de svåraste bitarna med att vara lärare och vi gör det väldigt noggrant. Vi ger feedback i de fall där vi upplever att det är meningsfullt men om du upplever att något inte gick rätt till så går det bra att höra av dig.