

TENTAMEN
För kursen
Introduktion till Programmering
TIG003

DATUM: 2021-01-18
TID: 09:00-13:00
PLATS: ONLINE

Kursansvar: Alan B. Carlson

Förfrågningar: 031-786 2786

Max poäng: 60

Betygsskala: G 36 p, VG 50 p

Rättning: Alan
Ingela
Andreas

Fråga nr: 1-5
7-9
6, 10-12

1) Satslogik vs Predikatlogik (3p)

Förklara och exemplifiera skillnaden mellan satslogik och predikatlogik. Använd och referera till kursmaterialet (boken och föreläsningarna).

2) Mängddefinitioner (2p)

Antag följande definitionsområde:

$$U = \{0, 1, 2, 3, 4\}$$

Visa vilka mängder (vilka element från definitionsområdet) representeras av följande uttryck:

- $\exists y (x * y = x)$
- $\exists y (x - y > 0)$

3) Logisk sanning, logisk falskhet, logisk kontingens (6p)

Visa med sanningsvärdestabeller om följande satser är logiskt sanna, falska eller kontingenta (dvs varken logiskt sann eller logiskt falsk):

- $\neg P \rightarrow (P \rightarrow Q)$
- $(P \leftrightarrow Q) \leftrightarrow \neg (P \leftrightarrow \neg Q)$
- $(P \vee Q \rightarrow R) \leftrightarrow ((P \rightarrow R) \& (Q \rightarrow R))$

4) Logisk konsekvens (4p)

Visa med sanningsvärdestabeller om slutsaten (ii) är en logisk konsekvens av premisserna (i):

- (i) $(P \rightarrow Q) \rightarrow P$, $P \rightarrow Q$ (ii) P
- (i) $Q \rightarrow \neg R$, $\neg P \rightarrow R$ (ii) $\neg P \rightarrow \neg Q$

5) Predikatlogiks semantik (3p)

Gör om följande påstående till predikatlogiska satser:

- Det finns en stad mellan Boxholm och Lund.
- Endast Skottar som dricker whiskey äter haggis.
- Alla hundar förutom chihuahuor gillar kyla.

6) Satslogiska härledningar (6p)

Situationen: Om uppror hotar är landet inte säkert. Kungen reser om och endast om uppror inte hotar. Landet är säkert. Alltså reser kungen. Kan formaliseras logiskt som:

$$\begin{aligned} \text{Uppror hotar} &= p \\ \text{Landet är säkert} &= q \\ \text{Kungen reser} &= r \\ p \rightarrow \neg q, r \leftrightarrow \neg p, q &\therefore r \end{aligned}$$

Visa med hjälp av satslogisk härledning att detta är korrekt.

7) Inmatning och typovandling (2+2+2p)

a. Ta in två tal via tangentbordet och skriv ut summan av dem.

b. Läs in text från användaren och skriv ut om talet är ett heltal med hjälp av `isdigit()`

c. Vad blir utskriften av följande rad och varför?

```
print(int(5 / 2.0))
```

8) Loopar (1+2+3p)

a) Följande kod är tänkt att användas för att loopa tills användaren skriver q, men något gör att koden får ett error och inte ens går in i loopen. Ge ett förslag för hur man kan få koden att fungera?

```
while svar != 'q':  
    print('Du skrev inte q')  
    svar = input('Skriv in q för att gå vidare')
```

b) Skapa och skriv ut en rad med ett antal stjärnor som kan ändras med en variabel. Ex ska '*****' skrivas ut om variabeln är satt till 5. Tips: Du kan öka en sträng s med s += "..."

c. Skapa en loop som skriver ut en mer stjärna per rad, skriv ut tio rader

```
*  
**  
***
```

9) Villkorssatser (6p)

Roboten Arduinna ska ut på en springtur i skogen efter att ha fått nya sensorer som kan upptäcka både höga och låga hinder samt några balla trick för att ta sig fram. Det saknas fortfarande kod för när Arduinna ska göra vad beroende på vad som upptäcks.

Om inga hinder finns springer Arduinna helst vanligt (kommando "spring"), rullar när det finns höga hinder som grenar ("rulla") och hoppar över stenar eller andra låga smådjur ("hoppa"). Ifall det samtidigt finns både höga och låga hinder krävs ett lite läskigare "snurrohopp" för att undvika båda, men vem vill trampa på en grävling?

Uppgift: Lägg till en eller flera if-satser inuti loopen för att hantera situationerna. Skriv ut vilket kommando som ska göras när, ex print("hoppa"). **OBS:** Det får endast ges ett kommando per situation/varv i loopen!

Du får:

2p om låga hinder hoppas över

2p om höga hinder rullas under

2p om ett snurrohopp görs för att undvika både lågt och högt hinder i en rörelse

Given kod:

```
import random  
  
# ger nya bool-värden för om det finns hinder framför  
def look_ahead():  
    low = random.choice([True, False])  
    high = random.choice([True, False])  
    return low, high  
  
energi = 10  
# fortsatt tills batteriet tar slut  
while True:  
    energi -= 1  
    if energi <= 0:  
        break  
    # nya bool-värden för varje varv i loopen  
    low_obstacle, high_obstacle = look_ahead()
```

10) Funktioner (1+2+3p)

a) Skriv en funktion som jämför två godtyckliga heltal, `num1` och `num2`, och skriver ut vilket som är störst (eller om de är lika). Komplettera med kod som anropar denna funktion med två tal som tagits in från användaren.

b) Skriv en funktion som adderar tre godtyckliga heltal (den ska alltså fungera med vilka tre heltal som helst). Komplettera med kod som anropar funktionen med tre tal (som tagits in från användaren) och lagrar resultatet i en variabel.

c) Skriv en funktion som tar in för- och efternamn som separata parametrar och returnerar desamma omvandlade till versaler. För att åstadkomma detta används den inbyggda funktionen `upper()`. Ex: om man har variabeln `myname = "Andreas"` så kan man omvandla den till versaler med:

```
myname = myname.upper()
```

11) Listor (6p)

En webbshop har lagrat sina produkter i en lista enligt formatet:

```
products = ["MacBook Pro", 126999, "iPhone SE", 4999, "iPad Pro", 11700]
```

Man vill nu separera produktnamnen från priserna i två separata listor. Skriv kod som åstadkommer detta.

12) Dictionaries (4 + 2p)

a) Utvecklaren hos webbshopen i föregående uppgift kommer nu på att det hade varit bättre att lagra produkterna med tillhörande priser i ett dictionary. Skriv kod som gör just detta.

b) Lagg nu till funktionalitet för att enkelt lägga till nya produkter med tillhörande priser som läggs in i dictionaryt. Denna information ska tas in från användaren. Skriv kod som gör just detta.