



STUDENT

0009-SCJ

TENTAMEN

TIG003 Tentamen

Kurskod	--
Bedömningsform	DO
Starttid	24.08.2023 13:30
Sluttid	24.08.2023 17:30
Bedömningsfrist	--
PDF skapad	13.11.2023 14:52
Skapad av	Lyudmila Vafaeva

i Information

TENTAMEN

TIG003: Introduktion till programmering

DATUM: 2023-08-24

TID: 13:30 - 17:30

PLATS: Reutersgatan 2C

Ansvarig: Andreas Chatzopoulos

Förfrågningar: 076-6183754

BETYGSGRÄNSER

Max poäng: 60 p

G: 30 p

VG: 45 p

Inga hjälpmedel.

1 Variabler och datatyper

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
age = 28
print(type(age))
```

```
b)
1) float
2) int
3) str
4) bool
```

Besvarad.

2 Loopar

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
telefonnummer = "08675309"
i = 0
for _ in range(len(telefonnummer):
    print(telefonnummer[i])
    i += 1

b)
heltal = input("Skriv ett antal heltal: ")
lista = list(heltal)

antal = 0
for num in lista:
    antal += 1
print(antal)

c)
saldo = 0
while True:
    val = input("Välj: 1)Insättning 2)Avsluta ")
    if val == 1:
        saldo = float(input("Ditt nuvarande saldo: "))
        insättning = float(input("Insättning i SEK: "))
        saldo += insättning
    elif val == 2:
        print(f"Saldo:{saldo:.2f}") #Saldo kommer vara 0 om anv. inte satt in
        #några pengar
        break
    else:
        print("Välj alt. 1 eller alt. 2")
```

Besvarad.

3 Villkor

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
#Tolkar över 18 som 18 eller äldre, eftersom det är definitionen av att vara
#myndig
age = int(input("Din ålder: "))

if age >= 18:
    print("Myndig")
else:
    print("Inte myndig")

b)
blandad_lista = ["Amanda", 1995, "Jacob", 1996, "Sara", 2002]

for element in blandad_lista:
    if isinstance(element, str):
        print(element)

c)
#Tolkar äldre än 65 som att åldern måste vara minst 66 för rabatt
age = int(input("Din ålder: "))
price = float(input("Priset på biobiljetten: "))

if age > 65:
    disc_price = price - (price * 0.15)
    print(f"Ditt pris: {disc_price:.2f}. Du har fått 15% i rabatt")
else:
    years_left = 66 - age
    print(f"Det är {years_left} år kvar tills du får rabatt")
```

Besvarad.

4 Strängar

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
#Tolkar blanksteget mellan orden i strängen som ett mellanslag i samtliga uppg.
Printar: an/or

b)
Printar: 8 ball

c)
Printar: no sirapflaska

d)
ny_straeng = straeng[3:7] + straeng[27:31]
```

Besvarad.

5 Dictionaries

Skriv in ditt svar här

```
a)
for namn in names:
    x = salaryDict[namn]
    print(f"Lön för {namn}:{x}")

#Om jag inte redan visste att namnen som finns i namnlistan även finns i dict
#hade jag undersökt om namnet finns i dict först för att inte få error när jag
#försöker plocka ut värdet
#Ex.
#for namn in names:
#    #if namn in salaryDict:
#        #x = salaryDict[namn]
#        #print(f"Lön för {namn}:{x}")

b)
salaryDict["Henrik"] = 26800
salaryDict["Frida"] = 32540
```

Besvarad.

6 Tupler

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
produkter = ((MacBook Pro, 12499, 13), (iPhone, 9899, 24), (iPad, 13399, 6))

b)
Eftersom data i en tupel inte kan ändras kan det vara fördelaktigt att använda
en tupel när man då inte vill att data ska kunna ändras/råkas ändras. Tupler är
även fördelaktigt vid tillfällen då ordningen på datan spelar roll.
I fråga a) gör tupler det tydligt att produktnamn, pris och lagersaldo hör
samman som en enhet och att produktnamn alltid plockas ut med (som ett andra
index efter produkt[i]) index 0 (produkt[i][0]), pris med index 1 (produkt[i][1]
och lagersaldo med index 2 (produkt[i][2])).
```

Besvarad.

7 Listor

Ersätt med din uppgiftstext...

Skriv in ditt svar här

```
a)
1) print(places[2])
2) print(len(places))

b)
namn = input("Ange ett förnamn: ")
students.append(namn.capitalize())
students.sort()
print(students)

c)
for straeng in word_list:
    if len(straeng) % 2 != 0:
        print(straeng)

d)
over = []
under = []
for i in range(10):
    salary = float(input("Lön: "))
    if salary >= 50000:
        over.append(salary)
    elif salary < 50000:
        under.append(salary)

sorterad_over = sorted(over)
sorterad_under = sorted(under)

diffr = sorterad_over[-1] - sorterad_under[0]
print(f"Skillnaden mellan högsta och lägsta lönen: {diff}")
```

Besvarad.

8 Funktioner

Ersätt med din uppgiftstext...

Skriv in ditt svar här

a)

Om man har en funktion som ex. gör en beräkning eller på annat sätt bearbetar argumentet du skickar `in` i funktionen vid anrop vill du kunna spara det värdet för att ex. använda senare i ditt program eller göra något med det värdet på något sätt.

Om du har en funktion som inte returnerar något värde, så utför funktionen bara något. Det kan ex. vara att funktionen printar ut en sats eller något annat som du inte har för avsikt att använda igen i koden. Då behöver funktionen inte returnera något.

b)

I funktionen `def calc_age():` så har ingen parameter definierats innanför paranteser. När funktionen ska tilldela variabeln `age` värdet `= datetime.datetime.now().year - birth` så vet inte funktionen vilket värde `birth` ska tänkas vara.

I definitionen tar inte funktionen `in` någon parameter och vid anropet skickas `birth` med som parameter, vilket inte definierats för funktionen.

c)

d)

Definierar en funktion `add` som tar två parametrar (`num1, num2`)

Inuti funktionen försöker funktionen göra `num1` till `3`. Ett tal kan inte vara `=` ett annat tal

Inuti funktionen försöker funktionen göra `num2` till `4`. Ett tal kan inte vara `=` ett annat tal

Sedan är tanken att värdet `3 + 4` ska returneras

e)

En funktion kan inte förändra globala variabler om du inte uttryckligen skriver ut att du avser en `global` variabel. Utanför loopen kommer känner programmet bara till de globala variablerna. De beräkningar som görs inuti funktionen på lokala variabler kommer inte förändra de globala variablerna.

Besvarad.

9 Mängdteori

Skriv in ditt svar här

a)

Resultat: {1,2,3,4,5,6,7,8,9,10}

$\exists y (x = y = y)$

$x ? : (x = 1 = 1) \vee (x = 2 = 2) \vee (x = 3 = 3) \vee (x = 4 = 4) \vee (x = 5 = 5) \vee (x = 6 = 6) \vee (x = 7 = 7) \vee (x = 8 = 8) \vee (x = 9 = 9) \vee (x = 10 = 10)$

1 ? : true

2 ? : true

3 ? : true

... det kommer alltid finnas något x där $x = y = y$. Delmängden blir {1,2,3,4,5,6,7,8,9,10}

b)

Resultat: {10}

$\forall y (x \geq y)$

$x ? : (x \geq 1) \& (x \geq 2) \& (x \geq 3) \& (x \geq 4) \& (x \geq 5) \& (x \geq 6) \& (x \geq 7) \& (x \geq 8) \& (x \geq 9) \& (x \geq 10)$

1 ? : false

2 ? : false

3 ? : false

... endast när $x = 10$ kommer alla y vara mindre eller lika med x

Ord: 173

Besvarad.

10 Logiska begrepp

Skriv in ditt svar här

a)

Logisk ekvivalens innebär att påståendena/satserna delar samma sanningsvärdestabell.

Ex. Om P, så Q och Om Q så P

Ex. Figuren har 3 sidor är ekvivalent med Figuren är en triangel (de är båda sanna samtidigt)

b)

Disjunktion = eller/or/ v

Ex. $P \vee Q$ = påståendet är sant om P eller Q är sant (i detta fall även när P och Q är sanna samtidigt). Om antingen P eller Q (men inte båda samtidigt) ska vara sann krävs exklusiv disjunktion

Ex. Barbie är en bra film eller Oppenheimer är en bra film (så länge någon utav Barbie eller Oppenheimer är bra, så blir påståendet sant)

c)

Kvantifierare talar om för hur många av objektet som påståendet gäller.

Allkvantierare : för alla x

Existenskvantifierare : för något x

Kvantifierarna kan negeras

d)

Logiskt konnektiv kan vara negation, disjunktion, konjunktion, implikation, ekvivalens

De används för att skapa mer komplexa satser.

Ex. genom att använda konjunktion skapas ett logiskt uttryck som i fallet $P \& Q$ betyder att både P & Q måste vara sanna för att påståendet ska vara sant (alltså en mer komplex sats)

Ord: 184

Besvarad.

11 Logisk konsekvens

(Använd gärna editorns tabellfunktion för att skapa snygga tabeller.)

Skriv in ditt svar här

a) B är inte en logisk konsekvens av premiss ($A \wedge \neg A$). Det förekommer inga värderingar där ($A \wedge \neg A$) är sant och B sant. Premissen ($A \wedge \neg A$) är inte sann i något värdering.

A	B	$\neg A$	$(A \wedge \neg A)$	B
T	T	F	F	T
T	F	F	F	F
F	T	T	F	T
F	F	T	F	F

b) $\neg A$ är ingen logisk konsekvens av $A \rightarrow B$, $\neg B \rightarrow C$, $\neg C$. Det förekommer en värdering där samtliga premisser är sanna och $\neg A$ är sann MEN det förekommer också en värdering där samtliga premisser är sanna och $\neg A$ falsk. Därför är $\neg A$ ingen logisk konsekvens av premisserna.

A	B	C	$A \rightarrow B$	$\neg B \rightarrow C$	$\neg C$	$\neg A$
T	T	T	T	T	F	F
T	F	T	F	T	F	F
T	T	F	T	T	T	F!
T	F	F	F	F	T	F
F	T	T	T	T	F	T
F	T	F	T	T	T	T-
F	F	T	T	T	F	T
F	F	F	T	F	T	T

c) A är en logisk konsekvens av $A \vee B$, $B \rightarrow C$, $\neg C$. Det förekommer en värdering där premisserna är sanna och A falsk. Det förekommer ingen värdering där premisserna är sanna och A falsk. Därför är A är en logisk konsekvens av $A \vee B$, $B \rightarrow C$, $\neg C$

A	B	C	$A \vee B$	$B \rightarrow C$	$\neg C$	A
T	T	T	T	T	F	T
T	F	T	T	T	F	T
T	T	F	T	F	T	T
T	F	F	T	T	T	T-
F	T	T	T	T	F	F
F	F	T	F	T	F	F
F	T	F	T	F	T	F
F	F	F	F	T	T	F

Ord: 295

Besvarad.

12 Predikatlogisk semantik

Ersätt med din uppgiftstext...

Skriv in ditt svar här

Tolkar uppgiften som att $\text{gillar}(x,y) : x$ gillar y , samma struktur som $\text{älskar}(x, y) : x$ älskar y

a)

Alla gillar någon

b)

Någon gillas av alla

c)

Alla gillar sig själv

Ord: 32

Besvarad.