



## Försättsblad Prov Original

Kurskod	DT001G	Provkod	T115	Tentamensdatum	2019 - 01 - 09
Kursnamn	Datateknik GR (A), Informationsteknologi grundkurs				
Provnamn	Tentamen				
Ort	Sundsvall				
Termin					
Ämne					



Mittuniversitetet  
MID SWEDEN UNIVERSITY

# Tentamen

Informationsteknologi grundkurs GR (A), 7,5 hp, DT001G

Jimmy Åhlander

2019-01-09

Tid	5 timmar
Hjälpmedel	Inga
Maxpoäng	50
Krav för godkänt	För godkänt betyg på tentamen fordras betyg E, eller högre.

Preliminära betygsgränser	A	B	C	D	E
	90 %	80 %	70 %	60 %	50 %

**Instruktioner** Skriv **tydligt**. Oläsliga svar erhåller inga poäng även om svaret är korrekt.

Besvara uppgifterna på separata svarpapper. Skriv endast på framsidan av svarpappret. Påbörja varje uppgift på ett nytt blad. Svaret på en uppgift kan alltså sträcka sig över flera blad, men maximalt en uppgift, med underuppgifter, får besvaras på ett givet blad.

Använd inte penna med röd skrift.

Markera sidnummer och din kod längst upp till höger på *varje blad*, se nedanstående figur. Numrera bladen *i efterhand* så att uppgifterna kommer i en logisk ordningsföljd.

*Skriv aldrig ut ditt namn eller personnummer.*

Anonymitetskod: 000-00	Sidnr: 1
1. a)	

**Instruktioner till tentamensvakt** Försättsbladet och studentens svar skannas automatiskt och får *inte* häftas ihop. Studenten får efter avlagd tentamen medta dessa instruktioner med tillhörande uppgifter. Väljer de att inte göra detta ska dessa instruktioner med tillhörande uppgifter kasseras. Endast de ohäftade svarpappren och försättsbladet skickas tillbaka.

### Uppgift 1 (3p)

Rita av *nedanstående tabell*, och fyll i de saknade värdena. Värdet för varje rad skall vara samma, men i olika talbaser.

Binär	Decimal	Hexadecimal
1100 0011		
	25	
		25

### Uppgift 2 (6p)

Logiska grindar används i datorer för att utföra logiska operationer och ibland för mer sofistikerade syften som att skapa minnesceller. Logiska grindar representeras vanligtvis med symboler och namngivna ingångar samt utgångar.

Avbilda tre olika logiska grindar. Rita in deras symboler i din avbildning, och namnge dem. Skriv också ut sanningstabellen för varje grind.

### Uppgift 3 (4 x 2p)

Utför följande beräkningar och presentera slutresultaten för a), b) och c) med 6 bitar i binär form. *Visa dina beräkningar och motivera dina svar.*

- $1001\ 1110_2 + 11\ 1001_2$
- $23_{10} / 2_{10}$
- $-5_{10} - 1_{10}$
- Konvertera  $10_{10}$  till tre andra, valfria talbaser.

### Uppgift 4 (2 + 3p)

Hypotetiska kusiner kämpar ofta med datatekniska begrepp och enheter. Just nu har din kusin snöat in sig på Linux. Lyckligtvis kommer du till undsättning för att reda ut ett par begrepp och förklara hur operativsystemet fungerar.

- Förklara miljövariabeln `$PATH`.
- Visa för din kusin, steg-för-steg, hur du i terminalen skulle kunna navigera från roten till katalogen `/home/username` där du sedan skapar en mapp med namnet `test` som ska innehålla en fil med namnet `test.txt` i sin tur innehållandes resultatet från en körning av programmet `date`. *Förklara de kommandon och operatörer du använder.*

### Uppgift 5 (1 + 2 + 3p)

- Processorn sitter inte i direkt anknötning till primärminnet utan måste gå igenom ett speciellt ledningssystem. Vad kallas normalt detta ledningssystem?
- Von Neumanns flaskhals är relaterat till ledningssystemet i a) och hur det används. På vilket sätt?
- För att kommunicera med periferienheter måste processorn, utöver att gå genom ledningssystemet i a), även gå igenom en controller. Vad är en controller och vad har den för uppgift?

### Uppgift 6 ( 1,5 + 4,5p )

En processor, CPU, består huvudsakligen av tre enheter.

- Namnge de tre enheterna processorn består av.
- Beskriv kortfattat varje enhets uppgift.

### Uppgift 7 ( 4 x 1,5p )

Förklara kortfattat funktionen för följande UNIX-kommandon.

- ls
- more
- tail
- which

### Uppgift 8 ( 3p )

Vad kommer Pythonprogrammet nedan skriva ut? Förklara hur det fungerar.

Vad blir utdatat efter att du gjort samtliga av nedan angivna förändringar i koden?

- Ändra `i = 1` till `i = 9` på rad 1.
- Ändra `while ( i < 10 )` till `while ( i > 0 )` på rad 4.
- Ändrar `i += 1` till `i -= 1` på rad 10.

```
1. i = 1
2. while ( i < 10 ):
3.
4.     if( i%3 == 0 ):
5.         print( i )
6.
7.     else:
8.         print( i, end="\t" )
9.
10. i += 1
```

### Uppgift 9 (2 + 3p)

I Python är variabler centrala. Variabler kan anta olika datatyper. Vidare finns det regler för hur identifierare till bland annat variabler kan utformas.

- a) Ge exempel på tre olika datatyper i Python.
- b) Vilka av följande identifierare är godkända (syntaxmässigt, inte nödvändigtvis konventionsmässigt) som variabelnamn i Python? *Motivera varför för icke-godkända fall. Svara endast på deluppgifter du är säker på: +0.5p per korrekt deluppgift, 0p per obesvarad deluppgift, -0.5p per felaktigt besvarad deluppgift (minimum: 0p).*
  - i) `be5tv4r14ble`
  - ii) `random_variable`
  - iii) `def`
  - iv) `_specialVariable`
  - v) `10numbers`
  - vi) `my variable`

### Uppgift 10 (2p)

Översätt följande bitsekvens till en sträng. Strängen ska tolkas som ASCII, se bifogad ASCII-tabell. Använd en byte per tecken.

```
01001000011001010110101001110011011000010110111000100001
```

### Checklista

- ✓ Kryssa i alla påbörjade uppgifter i försättsbladet.
- ✓ Numrera bladen så att uppgifterna kommer i numerisk och logisk ordningsföljd.

## Delar av ASCII-tabell, ISO 646, 7 bitar

Binärt	Symbol	Binärt	Symbol
010 0000	Space	100 1111	O
010 0001	!	101 0000	P
010 0010	"	101 0001	Q
010 0011	#	101 0010	R
010 0100	\$	101 0011	S
010 0101	%	101 0100	T
010 0110	&	101 0101	U
010 0111	'	101 0110	V
010 1000	(	101 0111	W
010 1001	)	101 1000	X
010 1010	*	101 1001	Y
010 1011	+	101 1010	Z
010 1100	,	101 1011	[
010 1101	-	101 1100	\
010 1110	.	101 1101	]
010 1111	/	101 1110	^
011 0000	0	101 1111	~
011 0001	1	110 0000	
011 0010	2	110 0001	a
011 0011	3	110 0010	b
011 0100	4	110 0011	c
011 0101	5	110 0100	d
011 0110	6	110 0101	e
011 0111	7	110 0110	f
011 1000	8	110 0111	g
011 1001	9	110 1000	h
011 1010	:	110 1001	i
011 1011	;	110 1010	j
011 1100	<	110 1011	k
011 1101	=	110 1100	l
011 1110	>	110 1101	m
011 1111	?	110 1110	n
100 0000	@	110 1111	o
100 0001	A	111 0000	p
100 0010	B	111 0001	q
100 0011	C	111 0010	r
100 0100	D	111 0011	s
100 0101	E	111 0100	t
100 0110	F	111 0101	u
100 0111	G	111 0110	v
100 1000	H	111 0111	w
100 1001	I	111 1000	x
100 1010	J	111 1001	y
100 1011	K	111 1010	z
100 1100	L	111 1011	{
100 1101	M	111 1100	
100 1110	N	111 1101	}
		111 1110	~