



## Försättsblad Prov Original

Kurskod	Provkod	Tentamensdatum
D T 1 5 5 G	T 1 0 1	2 0 1 8 - 0 8 - 2 4
Kursnamn	Datateknik GR (A), Grundläggande datavetenskap	
Provnamn	Skriftlig tentamen	
Ort	Sundsvall	
Termin	H18	
Ämne	Datateknik	

————— **Tentamen** —————  
**Datateknik GR (A) Grundläggande datavetenskap**  
**6hp: DT027G, 7,5 hp: DT155G och DT013G**  
————— **2018-08-24** —————

**Tid:** 5 timmar  
**Hjälpmedel:** inga  
**Maxpoäng:** 60

**Krav för godkänt:** Tentamen består av 4 avsnitt.  
För godkänt på tentamen krävs godkänt på varje avsnitt i tentamen, var för sig.  
Preliminär gräns för godkänt är 50 % på varje avsnitt.

<b>Preliminära betygsgränser:</b>	E 50 %*)	3 50 %	G 50 %
	D 60 %*)	4 67 %	VG 75 %
	C 70 %*)	5 83 %	
	B 80 %*)		
	A 90 %*)		

\*) Betygen A–E som ges på tentamen, gäller bara för tentamen. Dessa behöver inte stämma överens med totalbetyget för kursen. Olika kurskoder och versioner av kursplan innebär olika sätt att beräkna betyg. Detta står i så fall i kursplanen.

Observera.

- Skriv enkelt och **tydligt**, och notera alla (lämpliga) antaganden.
- **Skriv bara på ena sidan.**
- Du får inte använda penna med röd skrift!
- Lämna utrymme på varje blad för rättande lärare att notera poäng och eventuella kommentarer.

## Avsnitt 1

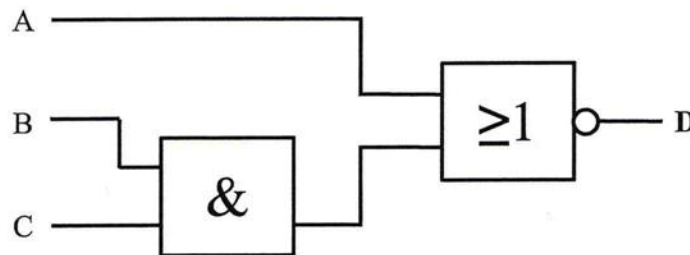
### Uppgift 1 (6p)

Rita av nedanstående bild, och fyll i de saknade värdena. Värdet inom varje rad ska vara samma, men i olika talbaser.

Bin	Dec	Hex	Oct
1001 1001			
	231		
		31	
			231

### Uppgift 2 (4p)

Skapa sanningstabellen för nedanstående koppling (Europeisk standard).



### Uppgift 3 (2p)

Värdet  $-54_{10}$  är lagrat på 2-komplementform i en byte. Skriv detta värde i hexadecimal form.

#### Uppgift 4 (3p)

För var och en, ange vilket begrepp som beskrivs.

- Den del av ett operativsystem som kommunicerar med användarna.
- Här ligger de data som ska hanteras av ALU.
- En dynamisk aktivitet där egenskaperna ändras med tiden när programmet exekveras.
- Om en eller flera av ingångarna är sann så blir utgången sann.
- Pekar ut nästa instruktion som ska hämtas för exekvering av processorn.
- Denna bit är värd minst.

#### Uppgift 5 (5p)

I ett program, vill vi kunna beräkna medelvärdet av 4 heltal. Talen finns lagrade på adresserna 84-87, och resultatet skall läggas i B3. Vi får tyvärr stå ut med att resultatet inte klarar decimaler.

Det är en enkel dator som kan köra maskinkod motsvarande den andra laborationen. Du har tillgång till maximalt 16 generella register och 256 minnesceller. Storleken på register och minnesceller är 8 bitar.

Tillgängliga instruktioner:

Op-kod	Operand	Beskrivning
1	RXY	Ladda (LOAD) registret R med bitmönstret som finns i minnescellen med adressen XY.
3	RXY	Spara (STORE) bitmönstret som finns i register R i minnescellen med adressen XY.
5	RST	Addera (ADD) bitmönstren i registren S och T. Lägg resultatet i register R. Bitmönstren antas vara kodade enligt 2-komplementmetoden.
D	R0X	Skifta bitmönstret i register R en bit åt vänster X gånger. Bit 1 (MSB) behåller sitt värde. Det hål som uppstår fylls med en 0:a
E	R0X	Skifta bitmönstret i register R en bit åt höger X gånger. Bit 1 (MSB) behåller sitt värde. Den bit som "faller över kanten" ersätts inte.
C	000	Stoppa (HALT) programkörningen.

## Avsnitt 2

### Uppgift 6 (4p)

För var och en, ange vilket begrepp som beskrivs.

- a) Detta lager ansvarar för transporten till nästa punkt på vägen till mottagaren.
- b) Den tittar på IP-adresser för att avgöra vilken port som inkommande trafik ska skickas vidare på.
- c) Datakommunikation sker normalt mellan dessa, inte mellan datorer.
- d) Detta protokoll sätter ihop inkomna paket, och skickar vidare uppåt. Det bryr sig inte om saknade paket, de hoppas över.
- e) Ett språk för att beskriva hur webbsidor ska presenteras.
- f) Internt nätverk byggt med Internet-teknik.
- g) Protokoll för överföring av webbsidor.
- h) Mäts i bitar per sekund (bps eller b/s).

### Uppgift 7 (4p)

Inom datakommunikation finns något vi kallar paket.

Vilka är huvuddelarna i ett paket?

Ge även en kort beskrivning av dessa delar.

### Uppgift 8 (2p)

Det tar 12 sekunder att överföra en fil med storleken 45 MB.

- Hur stor är genomsnittliga överföringshastigheten?

Svara i enheten kbps.

Redovisa dina beräkningar. *(Ingen redovisning => 0 poäng)*

### Avsnitt 3

#### Uppgift 9 (5p)

Skriv pseudokod för ett enkelt program där användaren kan mata in *valfritt antal* tal och som sedan presenterar det lägsta av dessa tal.

Användaren ska börja med att ange hur många tal hen vill mata in.

Glöm inte att en användare behöver lämpliga ledtexter för att förstå både in- och utdata.

Exempel på programdialog (Fet stil indikerar inmatade värden från användaren.):

```
Detta program skriver ut det lägsta talet av ett antal inmatade tal.  
Ange hur många tal du vill mata in: 4
```

```
Mata in ett tal: 2  
Mata in ett tal: -3  
Mata in ett tal: 22  
Mata in ett tal: 0
```

```
Det lägsta talet är: -3
```

Observera att du ska skriva kod, och använda det skrivsätt för normal pseudokod du lärt dig under kursen. Flödesschema ger 0 poäng.

#### Uppgift 10 (6p)

Vad skrivs ut från vart och ett av följande python3-program?

Observera att radbyten och blanktecken med flera ska tydligt synas i utskriften. (Tänk på alla vita tecken.)

Radmatning skriver du med symbolen `←`, blanktecken med `==`.

a.)	b.)	c.)
<pre>j = 7 while( j &gt; 3 ):     i = 2     while( i &lt; j ):         print( j, end="" )         i += 2     print()     j -= 1</pre>	<pre>b = 100 a = 4  a *= a  print( a, b )  c = a + b a = c - a b = c - b  print( a, b )</pre>	<pre>a = 23 b = 7 c = 5  print( a%b )  print( c // a )  c = c * c  print( c + a // b ) a = b + c  print( 27843697 % 2 )</pre>

### Uppgift 11 (4p)

För var och en, ange vilket begrepp som beskrivs.

- Ett av resultaten från denna är användarens kravspecifikation.
- En uppsättning satsar som utförs om t.ex. ett logiskt villkor är sant.
- En ordnad uppsättning entydiga anvisningar som utförs stegvis och definierar en avslutande process.
- Enligt denna tenderar fel att klumpa ihop sig.
- Detta steg ger en uppdelning av programmet i moduler som är oberoende utbytbara.
- Det språk som datorns processor förstår.
- Ett program för att skriva källkod i.
- En algoritmstruktur i en annan algoritmstruktur.

### Uppgift 12 (4p)

Översätt följande pseudokod till ett flödesschema:

```
j = 7
Så länge som( j > 3 )
  i = 2
  Så länge som( i < j )
    Skriv ut( j )
    Öka i med 2
  Skriv ut ny rad
  Minska j med 1
```

Det ska vara ett tydligt flödesschema (flödesdiagram) som vi kan följa. Det ska tydligt visa att du kan identifiera och översätta iteration, sekvens och selektion.

*Observera att det är ett flödesschema du ska rita, och du ska använda de symboler du lärt dig under kursen. Programkod eller pseudokod ger 0 poäng.*

## Avsnitt 4

### Uppgift 13 (3p)

Tabellen nedan representerar en stack som lagras i en kontinuerlig lista (fält, array).

- Vilket värde returneras vid en pop-instruktion om basen har adressen 13 och toppen 18?
- Vilket värde har stackpekaren efter denna operation?
- Vi fortsätter efter detta med en push-instruktion med invärdet 15. På vilken adress hamnar det värdet?
- Vilket värde har stackpekaren efter denna operation?

<i>Adress</i>	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
<i>Innehåll</i>	B	D	R	k	3	F	P	G	z	x	ç	T	20	B	5

### Uppgift 14 (3p)

I en relationsdatabas används begreppen *relation*, *tupel* och *attribut*.

Rita en lämplig skiss och markera dessa begrepp i skissen.

### Uppgift 15 (3p)

För var och en, ange vilket begrepp som beskrivs.

- "Riktighet/pålitlighet" i databaser.
- Överskott av data som inte ger mer information.
- En variabel som innehåller en minnesadress.
- Två av denna i samma relation kan inte vara identiska.
- En filtyp där "Minsta enheten av bitar" kodas/tolkas som tecken (char) enligt någon bestämd teckenkodning.
- En typisk egenskap för denna datastruktur är LIFO.

### Uppgift 16 (2p)

Översätt följande tecken-sträng till en bitsekvens. Det ska tolkas med ASCII. Använd en byte per tecken.

Gauge#5



## 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	~