Försättsblad Prov Original

<table>
<thead>
<tr>
<th>Kurskod</th>
<th>Provkod</th>
<th>Tentamensdatum</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT027G</td>
<td>T108</td>
<td>2019-01-10</td>
</tr>
</tbody>
</table>

Kursnamn: Datateknik GR (A), Grundläggande datavetenskap
Provnamn: Skriftlig tentamen
Ort: Sundsvall
Termin: 
Åmne:
Tentamen

Datateknik GR (A) Grundläggande datavetenskap
6hp: DT027G, 7.5 hp: DT155G

2019-01-10

Tid: 5 timmar
Hjälpmedel: inga
Maxpoäng: 60

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

Preliminära betygsgränser:

<table>
<thead>
<tr>
<th>Betyg</th>
<th>E 50 %*)</th>
<th>D 60 %*)</th>
<th>C 70 %*)</th>
<th>B 80 %*)</th>
<th>A 90 %*)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 50 %</td>
<td>4 67 %</td>
<td>5 83 %</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G 50 %</td>
<td>VG 75 %</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


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.

<table>
<thead>
<tr>
<th>Bin</th>
<th>Dec</th>
<th>Hex</th>
<th>Oct</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101 0110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td></td>
<td>9F</td>
<td>125</td>
</tr>
</tbody>
</table>

Uppgift 2 (4p)
Skapa sanningstabellen för nedanstående koppling (Europeisk standard).

[Diagram of a logic gate]

Uppgift 3 (2p)
Värdet -67₁₀ är lagrat på 2-komplementform i en byte. Skriv innehållet i denna byte på hexadecimal form.
Uppgift 4  (3p)
För var och en, ange vilket begrepp som beskrivs.

a) Innehåller den instruktion som ska exekveras av processorn.

b) Om minst en ingång är sann så blir utgången falsk.

c) Här läggs det ut vilken plats i primärmninet som det ska läsas från
eller skrivas till.

d) Flera program och användare delar på samma processor. Processerna
("de startade programmen") exekveras en i taget med snabba
växlingar.

e) En dynamisk aktivitet där egenskaperna ändras med tiden när
programmet exekveras.

f) Denna bit är värdd minst.

Uppgift 5  (5p)
I ett program vill vi kunna beräkna produkten av ett tal gånger 5.
Exempelvis om talet är 9 ska produkten bli 45, om talet är -12 ska produkten
bli -60. Talet finns lagrat i 2-komplementform i minnet på adress AA.
Produkten (svaret) ska läggas på adress 77.

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.

*Varje vald instruktion får användas max två gånger i din kod.*

Tillgängliga instruktioner:

<table>
<thead>
<tr>
<th>Op-kod</th>
<th>Operand</th>
<th>Beskrivning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RXY</td>
<td>Ladda (LOAD) registret R med bitmönstret som finns i minnescellen med adressen XY.</td>
</tr>
<tr>
<td>3</td>
<td>RXY</td>
<td>Spara (STORE) bitmönstret som finns i register R i minnescellen med adressen XY.</td>
</tr>
<tr>
<td>5</td>
<td>RST</td>
<td>Addera (ADD) bitmönstren i registren S och T. Lägg resultatet i register R. Bitmönstren antas vara kodade enligt 2-komplementmetoden.</td>
</tr>
<tr>
<td>D</td>
<td>R0X</td>
<td>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:</td>
</tr>
<tr>
<td>E</td>
<td>R0X</td>
<td>Skifta bitmönstret i register R en bit åt höger X gånger. Bit 1 (MSB) behåller sitt värde. Den bit som &quot;faller över kanten&quot; ersätts inte.</td>
</tr>
<tr>
<td>C</td>
<td>000</td>
<td>Stoppa (HALT) programköringen.</td>
</tr>
</tbody>
</table>
Avsnitt 2

Uppgift 6  (4p)
För var och en, ange vilket begrepp som beskrivs.

a) Består av ett huvud (header) och nyttolast (payload).

b) Den tittar på MAC-adresser för att avgöra vilken port som inkommande trafik ska skickas vidare på.

c) En exekverande process som tar initiativ till kommunikation.

d) Detta protokoll sätter ihop inkomna paket, och skickar vidare uppåt. Om ett paket saknas, begär det en ny kopia av paketet för att kunna sätta ihop paketen rätt.


f) En uppsättning regler som beskriver hur kommunikationen ska gå till.

g) Det vanligaste sidbeschreibungsspråket på webben.

a) Datakommunikation sker normalt mellan dessa, inte mellan datorer.

Uppgift 7  (4p)
Namnge och ge en kort beskrivning av varje lager i TCP/IP-stacken. De måste anges i rätt ordning.

Uppgift 8  (2p)
Hur lång tid tar det att överföra en fil med storleken 25 MB om den genomsnittliga överföringshastigheten är 5 kbps?
Svara i sekunder.

Redovisa dina beräkningar. (Ingen redovisning => 0 poäng)
Avsnitt 3

Uppgift 9 (6p)
Vad skrivs ut från vart och ett av följande python3-program?
Observera att radbyten och blanktecken med flera ska tydligt synas i utskrifterna. (Tänk på alla vita tecken.)

 Radmatning skriver du med symbolen ←, blanktecken med =.

<table>
<thead>
<tr>
<th>a)</th>
<th>b)</th>
<th>c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>b = 11</td>
<td>x = 6</td>
<td>a = 5</td>
</tr>
<tr>
<td>a = 2</td>
<td>while (x &gt; 2):</td>
<td>b = 12</td>
</tr>
<tr>
<td>print( b, a )</td>
<td>y = 1</td>
<td>c = 8</td>
</tr>
<tr>
<td>b = 7</td>
<td>while( x &gt; y ):</td>
<td>print( c )</td>
</tr>
<tr>
<td>c = a</td>
<td>if (x == 2*y):</td>
<td></td>
</tr>
<tr>
<td>a = b</td>
<td>print( y )</td>
<td>c = a * c</td>
</tr>
<tr>
<td>b = c</td>
<td>print( x )</td>
<td></td>
</tr>
<tr>
<td>c = 14</td>
<td>y += 1</td>
<td></td>
</tr>
<tr>
<td>print( b, a )</td>
<td>x = 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Uppgift 10 (4p)
För var och en, ange vilket begrepp som beskrivs.

a) Den kod som programmeraren skriver som blir ett program.
b) En uppsättning satser som utförs om t.ex. ett logiskt villkor är sant.
c) En ordnad uppsättning entydiga anvisningar som utförs stegvis och definierar en avslutande process.
d) Detta steg ska ge en uppdelning av programmet i moduler som är oberoende utbytbara.
e) Informellt och intuitivt språk för att beskriva algoritmer.
f) Ett hårdknodat värde, typ 3, 4.2 eller 'hej'.
g) Namngiven behållare för lagring ett värde som inte får ändras under exekvering.
h) En algoritmsstruktur där ett logiskt villkor styr om det ena eller det andra ska utföras.
Uppgift 11  (5p)
Skriv pseudokod för ett enkelt program som skriver ut alla tal mellan (och inklusive) två inmatade tal som är jämnt delbara med 2 angivna divisorer.

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

Detta program skriver ut alla tal mellan två värden som är jämnt delbara med två angivna divisorer.

Ange lägsta värdenet som testas: 5
Ange högsta värdenet som testas: 21
Ange 1:a divisorn: 2
Ange 2:a divisorn: 3

--- Resultat ---
6 är jämnt delbart med 2 och 3
12 är jämnt delbart med 2 och 3
18 är jämnt delbart med 2 och 3

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

Uppgift 12  (4p)
Översätt följande pseudokod till ett flödesschema:

\[ x = 6 \]
medans \( x > 2 \)
\[ y = 1 \]
medans \( x > y \)
om \( x = 2 \cdot y \)
Skriv \( y \)
Skriv \( x \)
Öka \( y \) med 1
Minska \( x \) med 2


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 (2p)
Översätt följande bitsekvens till en tecken-sträng. Det ska tolkas med
ASCII. Använd minsta antal bitar per tecken.

11000010111101011001011110000100000101101111100111011101

Uppgift 14 (3p)
För var och en, ange vilket begrepp som beskrivs.

a) Kan beskrivas som ”Riktighet/pålitlighet” i databaser.
b) Används för att skapa statistiska/continuerliga listor.
c) En typisk egenskap för denna datastruktur är LIFO.
d) Överskott av data som inte ger mer information.
e) Huvudmålen är att samordna, strukturerat och dokumentera
   information som en gemensam resurs.
f) Två av denna i samma relation kan inte vara identiska.

Uppgift 15 (3p)
Nedan ser vi en bit i minnet med olika värden lagrade. De olika
minnescellernas adresser står längst upp till vänster i respektive cell. I detta
minne finns det en länkad lista. Varje element i listan består av två byte,
inklusive pekare. Listans huvud har adress BA.

Uppgift
Ange värdena som finns lagrade i listan i rätt ordning.

<table>
<thead>
<tr>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>B7</th>
<th>B8</th>
<th>B9</th>
<th>RA</th>
<th>RB</th>
<th>RC</th>
<th>BD</th>
<th>BE</th>
<th>BF</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>B4</td>
<td>A</td>
<td>NIL</td>
<td>BC</td>
<td>B6</td>
<td>BB</td>
<td>BC</td>
<td>5</td>
<td>B8</td>
<td>BA</td>
<td>B6</td>
</tr>
</tbody>
</table>

Uppgift 16 (3p)

a) Vad är en textfil?

b) Vad kan skilja mellan olika typer av textfiler?
<table>
<thead>
<tr>
<th>Binärt</th>
<th>Symbol</th>
<th>Binärt</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>010 0000</td>
<td>Space</td>
<td>100 1111</td>
<td>O</td>
</tr>
<tr>
<td>010 0001</td>
<td>!</td>
<td>101 0000</td>
<td>P</td>
</tr>
<tr>
<td>010 0010</td>
<td>&quot;</td>
<td>101 0001</td>
<td>Q</td>
</tr>
<tr>
<td>010 0011</td>
<td>#</td>
<td>101 0010</td>
<td>R</td>
</tr>
<tr>
<td>010 0100</td>
<td>$</td>
<td>101 0011</td>
<td>S</td>
</tr>
<tr>
<td>010 0101</td>
<td>%</td>
<td>101 0100</td>
<td>T</td>
</tr>
<tr>
<td>010 0110</td>
<td>&amp;</td>
<td>101 0101</td>
<td>U</td>
</tr>
<tr>
<td>010 0111</td>
<td>(</td>
<td>101 0110</td>
<td>V</td>
</tr>
<tr>
<td>010 1000</td>
<td>)</td>
<td>101 0111</td>
<td>W</td>
</tr>
<tr>
<td>010 1001</td>
<td>*</td>
<td>101 1000</td>
<td>X</td>
</tr>
<tr>
<td>010 1010</td>
<td>+</td>
<td>101 1001</td>
<td>Y</td>
</tr>
<tr>
<td>010 1011</td>
<td>,</td>
<td>101 1010</td>
<td>Z</td>
</tr>
<tr>
<td>010 1100</td>
<td>;</td>
<td>101 1011</td>
<td>[</td>
</tr>
<tr>
<td>010 1101</td>
<td>-</td>
<td>101 1012</td>
<td>\</td>
</tr>
<tr>
<td>010 1110</td>
<td>.</td>
<td>101 1100</td>
<td>]</td>
</tr>
<tr>
<td>010 1111</td>
<td>/</td>
<td>101 1101</td>
<td>^</td>
</tr>
<tr>
<td>011 0000</td>
<td>0</td>
<td>101 1110</td>
<td>_</td>
</tr>
<tr>
<td>011 0001</td>
<td>1</td>
<td>110 0000</td>
<td>a</td>
</tr>
<tr>
<td>011 0010</td>
<td>2</td>
<td>110 0001</td>
<td>b</td>
</tr>
<tr>
<td>011 0011</td>
<td>3</td>
<td>110 0010</td>
<td>c</td>
</tr>
<tr>
<td>011 0100</td>
<td>4</td>
<td>110 0011</td>
<td>d</td>
</tr>
<tr>
<td>011 0101</td>
<td>5</td>
<td>110 0100</td>
<td>e</td>
</tr>
<tr>
<td>011 0110</td>
<td>6</td>
<td>110 0101</td>
<td>f</td>
</tr>
<tr>
<td>011 0111</td>
<td>7</td>
<td>110 0110</td>
<td>g</td>
</tr>
<tr>
<td>011 1000</td>
<td>8</td>
<td>110 0111</td>
<td>h</td>
</tr>
<tr>
<td>011 1001</td>
<td>9</td>
<td>110 1000</td>
<td>i</td>
</tr>
<tr>
<td>011 1010</td>
<td>;</td>
<td>110 1001</td>
<td>j</td>
</tr>
<tr>
<td>011 1011</td>
<td>&lt;</td>
<td>110 1010</td>
<td>k</td>
</tr>
<tr>
<td>011 1100</td>
<td>=</td>
<td>110 1011</td>
<td>l</td>
</tr>
<tr>
<td>011 1101</td>
<td>&gt;</td>
<td>110 1100</td>
<td>m</td>
</tr>
<tr>
<td>011 1110</td>
<td>?</td>
<td>110 1101</td>
<td>n</td>
</tr>
<tr>
<td>011 1111</td>
<td>@</td>
<td>110 1110</td>
<td>o</td>
</tr>
<tr>
<td>100 0000</td>
<td>#</td>
<td>110 1111</td>
<td>p</td>
</tr>
<tr>
<td>100 0001</td>
<td>A</td>
<td>111 0000</td>
<td>q</td>
</tr>
<tr>
<td>100 0010</td>
<td>B</td>
<td>111 0001</td>
<td>r</td>
</tr>
<tr>
<td>100 0011</td>
<td>C</td>
<td>111 0010</td>
<td>s</td>
</tr>
<tr>
<td>100 0100</td>
<td>D</td>
<td>111 0011</td>
<td>t</td>
</tr>
<tr>
<td>100 0101</td>
<td>E</td>
<td>111 0100</td>
<td>u</td>
</tr>
<tr>
<td>100 0110</td>
<td>F</td>
<td>111 0101</td>
<td>v</td>
</tr>
<tr>
<td>100 0111</td>
<td>G</td>
<td>111 0110</td>
<td>w</td>
</tr>
<tr>
<td>100 1000</td>
<td>H</td>
<td>111 0111</td>
<td>x</td>
</tr>
<tr>
<td>100 1001</td>
<td>I</td>
<td>111 1000</td>
<td>y</td>
</tr>
<tr>
<td>100 1010</td>
<td>J</td>
<td>111 1001</td>
<td>z</td>
</tr>
<tr>
<td>100 1011</td>
<td>K</td>
<td>111 1010</td>
<td>{</td>
</tr>
<tr>
<td>100 1100</td>
<td>L</td>
<td>111 1011</td>
<td>}</td>
</tr>
<tr>
<td>100 1101</td>
<td>M</td>
<td>111 1100</td>
<td>\</td>
</tr>
<tr>
<td>100 1110</td>
<td>N</td>
<td>111 1101</td>
<td>~</td>
</tr>
</tbody>
</table>