

# Tentamen, 2.5 högskolepoäng november 4, 2025, 8:15 - 12.15

*Du ska inte använda någon bok, bilder eller online-resurs. Du kan använda en fickräknare. Du får inte samarbeta med andra studenter via mobiltelefon eller chatt eller att prata. Misstänkta fall av samarbete och plagiering kommer att rapporteras och tentan anses ogiltig.*

*Du kan svara på svenska eller engelska, men engelska är att föredra. Oroa dig inte om din engelska inte är perfekt, den kommer inte att utvärderas, men svaren måste vara förståeliga.*

*Tentamen har 8 frågor med en sammanlagd poäng på 25. Krav för betyg 3 (godkänt) är **15** poäng, krav för betyg 4 är **19** poäng och krav för betyg 5 är **23** poäng.*

*Du borde svara med papper och penna. Se till att skriva numret på den fråga du svarar på. Svar som inte kan tolkas på grund av dålig handskrift eller dålig grammatik ger inga poäng.*

*För frågor, du kan ringa Dario till 0723756073.*

*Tips: läs igenom HELA tentamen först och lägg upp en strategi för i vilken ordning ni löser uppgifterna – tex de ni tycker är enkla först och de svåra i mån av tid.*

## Fråga 1 (4pt)

För varje kodexempel ange funktionen  $f(n)$  för vilken körningstiden är  $\Theta(f(n))$ .

Låt  $A$  vara en vektor (array) med  $n$  heltal.

(Notation:  $\leftarrow$  betecknar tilldelning,  $//$  är heltalsdivision)

Korrekt svar på en delfråga ger 1 poäng (4p)

a.

```
for i <- 0 to n-1 do
  A[i] <- A[i] + 1
```

b.

```
for i <- 0 to n-2 do
  for j <- i+1 to n-1 do
    A[i] <- i*j-1
```

c.

```
def compC(A, i, j) {
  if j-i < 4 then return min(A[i], A[i+1], A[i+2]) # Anta att min-funktionen finns och tar O(1) tid
  k1 <- (j-i)//3 + i # // är heltalsdivision
  k2 <- j - (j-i)//3
```

```
    return min(compC(A,i,k1), compC(k1+1, k2), compC(k2+1, j))  
}
```

Vad är komplexiteten för följande anrop?

```
x <- compC(A, 0, n-1)
```

d.

```
i <- 1  
while i < n do  
  A[i] <- 0  
  i <- 2*i
```

Answer:

a.  $\Theta(n)$  time

b.  $\Theta(n^2)$  time

c.  $T(n) = 3T(n/3) + \Theta(1) = \Theta(n)$  time. Master theorem!

d.  $\Theta(\log n)$  time

## Fråga 2 (4pt)

a. Varför behöver man omstrukturera ett binärt sökträd för att upprätthålla det balanserat ( $O(\log n)$  höjd)?

Ge exempel på en sekvens insättningar som leder till ett obalanserat träd! (2p)

b. En stack är en ADT med två operationer, push och pop. Förklara vad var och en av dem gör.

Vad är en lämplig datastruktur för att implementera en stack? Hur implementeras push och pop i din valda datastruktur? (2p)

Answer:

a. Annars kan en sekvens insättningar och/eller borttag leda till ett obalanserat träd. insättning av element i sorterad ordning i ett tomt träd, t.ex. 0, 1, 2, 3, ..., n

b. push sätter in ett element i stacken. Pop tar ut det senast insatta elementet som är kavr på stacken från den.

Lämplig datastruktur: t.ex. Enkellänkad lista.

## Fråga 3 (4pt)

a. Datastrukturen heap upprätthåller elementen enligt en regel. Vilken regel? (även kallad heapordning) (1p)

b. I en vektorimplementation av en (min) heap med åtta element befinner de sig i följande ordning

0 3 1 4 7 5 2 6

Visa hur vektorn ser ut efter var och en av tre successiva anrop till operationen deleteMin.  
(3p)

Answer:

a. För min-heap gäller att varje element är minst lika stor som sin förälder

b.

0 <- deleteMin 6 3 1 4 7 5 2 ... 1 3 2 4 7 5 6

1 <- deleteMin 6 3 2 4 7 5 ... 2 3 5 4 7 6

2 <- deleteMin 6 3 5 4 7 ... 3 4 5 6 7

#### Fråga 4 (3pt)

EN: Describe the way bootloading works on an ESP32. Describe each step including first and second stage bootloaders.

SE: Beskriv hur startladdning fungerar på en ESP32. Beskriv varje steg inklusive bootloaders i första och andra steget.

ANSWER:

First-stage bootloader: configures the access to the external flash memory and, if required, stores on it new data coming from the serial/USB port. Once finished, it accesses the flash memory (at address 0x1000) and loads and executes the second-stage bootloader.

Second-stage bootloader: reads the partition table at address 0x8000 and searches for app partitions. It decides which application must be executed based on the content of the *otadata* partition: if this partition is empty or doesn't exist, the bootloaded executes the application stored in the factory partition.

This allows to implement an over-the-air (OTA) application update process: the application code downloads the new version of your application (e.g. from web) and stores it in a new app partition. Once the upload is completed, the id of the partition is saved in *otadata* and the chip is rebooted; the bootloader will execute the new version.

#### Fråga 5 (2pt)

EN: Show how UML diagrams can be used for the 4+1 view architectural model. Fill in the following table by placing the UML diagrams from the list below into the right column.

SE: Visa hur UML-diagram kan användas för arkitekturmodellen med 4+1-vy. Fyll i följande tabell genom att placera UML-diagrammen från listan (se ner) i den rätta kolumnen.

<b>View</b>	Logical view	Process view	Development view	Physical view	Scenarios
<b>UML – Diagram</b>					

UML diagrams:

Class Diagram, Object Diagram, Activity Diagram, State Diagram, Timing Diagram, Sequence Diagram, Component diagram, Package diagram, Deployment diagram, Use case diagram.

**ANSWER:**

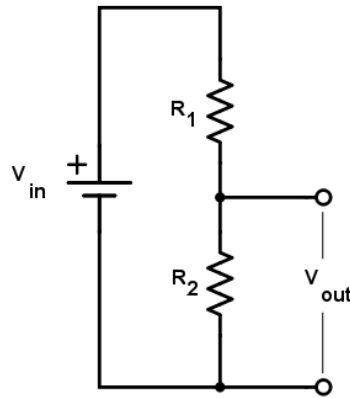
<b>View</b>	Logical view	Process view	Development view	Physical view	Scenarios
<b>UML – Diagram</b>	Class, Object Diagram	Activity, State, Timing, Sequence Diagram	Component, Package diagram	Deployment diagram	Use case diagram

### Fråga 6 (5pt)

EN:

You are developing a thermostat for a smart home environment. The thermostat works with batteries and should very little power in order for the batteries to last long.

You attach a thermistor to the board with the following circuit:



Where:

- $V_{in}$  is 3.3V
- $V_{out}$  is the voltage applied to the ADC converter of your microcontroller
- $R_1$  is a pullup resistor of 100 Ohm
- $R_2$  is the thermistor. According to the datasheet, the relationship between the resistance and the temperature is:

$$R = R_0 e^{\frac{K}{T}}$$

Where  $K$  and  $R_0$  are constants.

a) (3 pt)

Considering that the ADC uses 10 bits resolution (values from  $A_{min}=0$  to  $A_{max}=1023$ ), assuming that the ADC is perfectly linear, derive the function  $f$  that relates the integer generated by the ADC,  $A$ , with the temperature  $T = f(A)$ .

You program a first version of the firmware in this way:

- The board is in sleep mode until:
  - o A timer kicks in every 15 seconds. When woken up, the device measures the temperature of the room and decides if it needs to switch on the heaters or not
  - o The user presses a button to interact with the device.

b) (2pt)

Considering that the microcontroller operates at 250 MHz and that the timer has a 10 bits prescaler and a 32 bits compare register, find suitable values for both registers so that the microcontroller is woken up every 15 seconds.

Hint:  $2^{10} = 1024$  and  $2^{32} = 4294967296$

ANSWER:

A)

Using the voltage divider and assuming a linear relationship between the value measured at the ADC and the voltage applied to it, then  $R_2 = \frac{A R_1}{A_{max} - A}$ .

We also know from the datasheet that  $R_0 e^{\frac{K}{T}} = R_2 = \frac{A R_1}{A_{max} - A}$

Resolving for T:

$$\frac{A R_1}{R_0 (A_{max} - A)} = e^{\frac{K}{T}}$$
$$\frac{K}{T} = \log_e \left( \frac{A R_1}{R_0 (A_{max} - A)} \right)$$

Therefore:

$$T = K / \log_e \left( \frac{A R_1}{R_0 (A_{max} - A)} \right)$$

If we assume that Amax is 1023, R1 is 100, we get:

$$T = K / \log_e \left( \frac{A 100}{R_0 (1024 - A)} \right)$$

B)

15 seconds period is  $1/15 = 0.066$  Hz frequency. We need to find P and C so that  $250 \text{E}6 / (P C) = 1/15$ , that is:  $P C = 250 \text{E}6 \cdot 15 = 3750000000$

This gives us a hint that P C should be multiple of 10 (or 100, or 1000)

If we set the prescaler to close to P=1000:  $C = 250 \text{E}6 \cdot 15 / 1000 = 3750 000$  which fits within the 32 bits compare register.

### Fråga 7 (1 pt)

EN

The variable reg is an unsigned integer of 8 bits in C. How do you set the 3rd bit from the right to 1 in its value?

SE

Variabeln reg är ett heltal utan tecken på 8 bitar i C. Hur ställer du in den 3:e biten från höger till 1 av dess värde?

ANSWER:

`reg = reg | (1 << 2)`

### Fråga 8 (2 pt)

EN

- a) What happens when a function is declared with the keyword “extern” in C? (1pt)
- b) What happens when a variable is declared with the keyword “extern” in C? (1pt)

SE

- a) Vad händer när en funktion deklarerats med nyckelordet "extern" i C? (1 pkt)
- b) Vad händer när en variabel deklarerats med nyckelordet "extern" i C? (1 pkt)

### ANSWER:

- a) Nothing particular. An extern function is a function that is defined somewhere else (in another C file). Anytime a function is declared, it is by default extern (even if one doesn't use the extern keyword). In fact, functions are always global unless declared static.
- b) A variable declared as extern in a module (c-file) means that the variable is defined in another module. The memory space is thus reserved in another module, but the compiler knows the type even though it does not reserve any memory space. The variable is implicitly global.