Important: Carefully read through these instructions before starting with the exam.

Rules for the exam

- You are allowed to consult text books, lecture slides, lecture recordings, your own assignment reports and other material or tools given to you during the course while taking the exam.
- You **must** take the exam completely by yourself.
- You are **not** allowed to communicate with anyone (except a teacher, see below) during the exam, either in person, digitally, or otherwise.
- You are **not** allowed to sit right next to another person, or let someone else see your solutions.
- You are **not** allowed to ask any questions, online or otherwise.
- You are **not** allowed to post any solutions or hints anywhere.
- Any questions where you are asked to write an answer must be answered in your own words. You are **not** allowed to directly copy or naively paraphrase from course material.

Any infractions will be reported as per The Higher Education Ordinance (Högskoleförordningen), chapter 10.

Talk to a teacher

In case you are in need of some clarification about the exam, you will be able to contact one of the teachers during the exam using Zoom. For this, the Zoom room https://uu-se.zoom.us/j/68629229918 will be available from 09:00 to 10:00 and from 12:00 to 12:30 on the day of the exam. Only one student will be admitted at a time, so you may have to wait in a waiting room for your turn.

Note: Since the time where you can ask questions is limited. Make sure to read through the entire exam in order to spot any part that you find unclear before you start. We cannot give any hints about how to answer the questions, we can only attempt to clarify any ambiguities in the questions themselves.

About uploading solutions

Some of the questions you will answer directly here in Inspera by selecting an answer, filling in some text box etc. Other questions require you to upload a file with your solution.

For the latter type of question, the recommended way is to draw/write on paper and then scan or photograph the pages before uploading them. Alternatively, you may use some software of your choice to create drawings and text.

The only supported file types for your answers to these questions are **.jpg**, **.png** and **.pdf**. Due to technical limitations only a *single* file may be uploaded per question. If your answer spans several pages you must therefore merge them into a single file for the upload.

If you are using a smart phone to snap pictures of your written pages, then an app like Adobe Scan (Android, iOS) could be useful. You could also use free software like https://pdfsam.org/ to merge separate PDF files into a single file on your computer. Online services like https://www.ilovepdf.com/ or https://www.easepdf.com/ can also be used (but these can limit things like file size or number of operations).

Before uploading a file, make sure that both the handwriting and the image quality make it legible.
Make sure to write your name on the each uploaded page.

Good luck!

1 If a task set $\mathcal{T}$ **passes** a schedulability test that is **sufficient but not necessary**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative:

- It is schedulable
- Nothing
- It is unschedulable

If a task set $\mathcal{T}$ **fails** a schedulability test that is **sufficient but not necessary**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative

- It is unschedulable
- It is schedulable
- Nothing

If a task set $\mathcal{T}$ **passes** a schedulability test that is **necessary but not sufficient**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative

- Nothing
- It is schedulable
- It is unschedulable

If a task set $\mathcal{T}$ **fails** a schedulability test that is **necessary but not sufficient**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative

- Nothing
- It is unschedulable
- It is schedulable

If a task set $\mathcal{T}$ **passes** a schedulability test that is **exact**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative

- It is schedulable
- Nothing
- It is unschedulable

If a task set $\mathcal{T}$ **fails** a schedulability test that is **exact**, then we can conclude the following about the schedulability of $\mathcal{T}$.

Select one alternative

- Nothing
- It is unschedulable
- It is schedulable

Each correct answer gives 1 point. Each incorrectly answered or unanswered question gives -1 points. The minimum points this problem can give is zero (i.e., the total will never be negative).

Maximum marks: 6

2 The worst-case execution time (WCET) of a computer program is generally difficult to determine exactly. Broadly speaking, the WCET of a program can be estimated either with *static analysis* or with *testing*. Write answers to address both of the following.

1. Give an overview of a reasonable process to estimate the WCET of a program using static analysis. Describe the process clearly and intelligibly, and focus on the high-level ideas rather than the details.
2. Contrast *static analysis* to *testing* and describe the major differences in the two approaches, indicating benefits and drawbacks of each.

(Try to be as concise as possible. Maximum 600 words in total.)

Enter your answers here.

Words: 0/600

A maximum of 10 points can be given for this question.

Maximum marks: 10

3 Let $\mathcal{T}$ be a task set of n synchronous periodic tasks that are scheduled on *one* preemptive processor. Drag and drop the correct equations to their correct places below. Note that only four of the equations should be used.

(Each correct answer gives 2 points. Each incorrect answer gives 0 points.)

Help

$\forall \tau_i \in \mathcal{T} : D_i = T_i$

$\mathrm{U}(\mathcal{T}) \leq \frac{\beta n+1}{\beta+1}$, where $\beta = \left\lfloor \frac{1}{\mathrm{U_{max}}(\mathcal{T})} \right\rfloor$

$n! = \mathrm{lcm}_{\tau_i \in \mathcal{T}}\{T_i\}$

$\mathrm{U}(\mathcal{T}) \leq 1$

$\forall t > 0 : \sum_{\tau_i \in \mathcal{T}} \max\left(0, \left\lfloor \frac{t-D_i}{T_i} \right\rfloor + 1\right) \times C_i \leq t$

$\mathrm{U}(\mathcal{T}) \leq n(2^{\frac{1}{n}} - 1)$

$\forall \tau_i \in \mathcal{T} : R_i \leq D_i$, where R_i is the fixed point to $R_i = C_i + \sum_{\tau_j \in \mathrm{hp}(\tau_i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times C_j$

$\mathrm{U}(\mathcal{T}) = 1$

$\mathrm{U}(\mathcal{T}) \geq 0.69$

- If $\mathcal{T}$ has implicit deadlines and is scheduled by the Earliest Deadline First (EDF) scheduler, then the following condition is the most suitable exact schedulability test:

- If $\mathcal{T}$ has arbitrary deadlines and is scheduled by the Earliest Deadline First (EDF) scheduler, then the following condition is an exact schedulability test:

- If $\mathcal{T}$ has implicit deadlines and is scheduled by the Fixed-Priority (FP) scheduler with Rate-Monotonic (RM) priority ordering, then the following condition is a sufficient, but not necessary, schedulability test:

- If $\mathcal{T}$ has constrained deadlines and is scheduled by the Fixed-Priority (FP) scheduler, then the following condition is an exact schedulability test:

Maximum marks: 8

4 If tasks share mutually exclusive shared resources locked by semaphores or mutexes, then *deadlocks* can occur unless the scheduling is carefully managed, for example with a resource sharing protocol.

For this question, write and draw answers to *all* of the following sub-questions.

1. Give a concrete example of how a deadlock can occur without any resource sharing protocol. Show which tasks and semaphores/mutexes are involved in your example and clearly explain how the deadlock can occur. Draw a schedule where the deadlock happens.
2. If all the jobs are scheduled non-preemptively, can deadlocks still occur? Clearly explain your reasoning.
3. The Highest Locker Protocol (HLP) (sometimes called Immediate Priority Ceiling (IPC) or Immediate Priority Inheritance) is a resource sharing protocol. Clearly explain how it works and why it prevents deadlocks.

Upload your file here. Maximum one file.

The following file types are allowed: **.jpg,.png,.pdf** Maximum file size is **1 GB**

Select file to upload

It is recommended that you write/draw on paper and then scan or photograph what you have written before uploading your solution, but make sure that the result is legible. Alternatively, you may draw and write in some software of your choice. Supported file formats for the upload are .jpg, .png and .pdf. Note that you must upload a single file with your solutions to all the sub-questions.

This question can give a maximum of 11 points in total.

Maximum marks: 11

5 Write an Ada program as follows.

- The program should contain a single procedure named **Exam**.
- The procedure **Exam** should contain two tasks, named **T1** and **T2**.
- Both tasks should run forever.
- Task **T1** should print the text "*Task 1, checking in*" every 4 seconds, without any long-term drift.
- Task **T2** should print the text "*Task 2, checking in*" every 9 seconds, without any long-term drift.

Write your program here

Please use proper indentation. You don't need to include comments, but may do so if you wish.

A complete and error-free program (that can be compiled and run as-is if saved to a file named exam.adb), and that uses appropriate programming constructs, will give 10 points. There is also a possibility for partial credit if the program is incomplete, contains minor errors, or is in a poor programming style. The program should be valid Ada95.

(If you wish to, you are allowed to compile and test your program on your computer before submitting it here. For example, you can use the GNAT Ada compiler available on the virtual machine from the labs or download it from https://www.adacore.com/download. However, we cannot offer any software support during the exam.)

Maximum marks: 10

6 Let $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$ be a task set of three sporadic tasks with the following parameters.

	C	D	T
τ_1	5	20	20
τ_2	2	3	5
τ_3	3	7	11

Assume that $\mathcal{T}$ is scheduled by a Fixed-Priority scheduler with Deadline-Monotonic priority ordering on a single preemptive processor. Calculate the worst-case response times of the three tasks.

The worst-case response time of $\tau_1$is:

The worst-case response time of τ_2 is:

The worst-case response time of τ_3 is:

(Each correct answer gives 3 points. Each incorrect answer gives 0 points.)

Maximum marks: 9

7 Let $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$ be a task set of three synchronous periodic tasks with the following parameters.

	C	D	T
τ_1	1	3	3
τ_2	3	8	10
τ_3	4	12	15

The task set is scheduled with the Earliest Deadline First (EDF) scheduler on one preemptive processor.

1. Draw the schedule of $\mathcal{T}$ for the first 30 time units, and clearly indicate release times, deadlines, and job execution. Assume that every job needs to execute for exactly its WCET when drawing the schedule.
2. Then determine whether the task set $\mathcal{T}$ is EDF-schedulable or not. Write down your answer and *clearly* motivate why this is the case.

Upload the drawing of the schedule (1) and your answer (2) *as a single file*. It is recommended that you draw and write on paper and scan/photograph the page before uploading. Alternatively, you may draw the schedule and write your answer in the software of your choice. Supported file types for the upload are .jpg, .png and .pdf.

Upload your file here. Maximum one file.

The following file types are allowed: **.jpg,.png,.pdf** Maximum file size is **1 GB**

Select file to upload

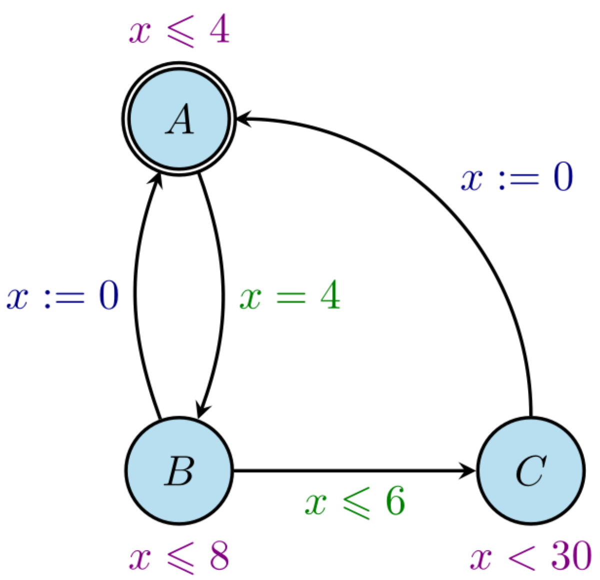
A maximum of 6 points can be given for this question.

Maximum marks: 6

8 The figures below show timed automata. Expressions with the binary operator := next to edges are updates (i.e., clock resets), while other expressions next to edges are guards or synchronizations. Expressions next to vertices are invariants. Labels inside vertices are location names. Doubly circled vertices are the initial locations.

Part 1

Below is the timed automaton **T1**, which uses one clock **x**.



Consider the following four questions.

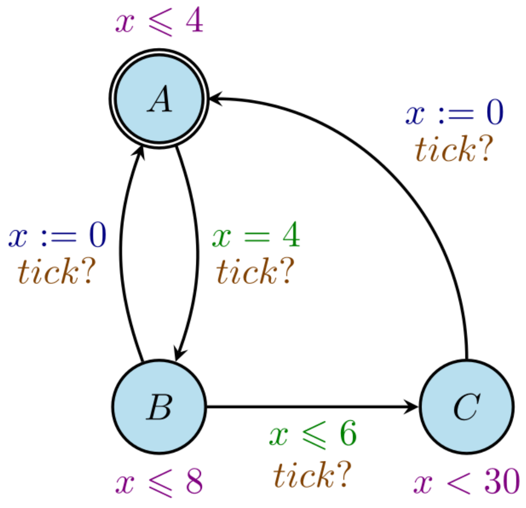
(a) Is location C reachable?
(b) Is location C reachable while x is equal to 3?
(c) Is it always the case that x is at least 3 when at location B?
(d) Can x ever have the value of exactly 10?

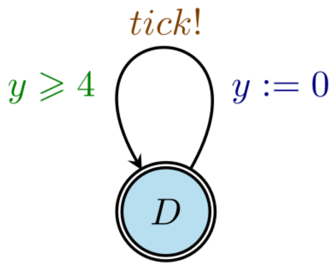
For each of the questions **(a)-(d)** above, do the following:

1. Express the question as a temporal logic query (as you would in Uppaal).
2. State whether the question/query is satisfied.
3. Clearly explain *why* it is satisfied or not.

Part 2

Below a new timed automaton **T2** (to the right) is added, with a new clock **y**. Both **T1** and **T2** now run concurrently. In addition, they synchronize over the channel **tick**, and every edge in **T1** has been extended to include a synchronization on this channel.





For this extended system, evaluate whether the four questions **(a)-(d)** from Part 1 are satisfied, and again clearly explain why.

Fill in your answers below. Mark your different answers as (1a), (1b), (1c), (1d), (2a), (2b), (2c) and (2d).

Fill in your answer here

Format

Words: 0

A total of 18 points can be given for this question.

(You should not have to use Uppaal to solve any of the above, but you are allowed to use Uppaal if you wish.)

Maximum marks: 18

9 Let $\mathcal{T}$ be a task set consisting of the following three sporadic tasks that are to be scheduled on **two** preemptive processors.

	C	D	T
τ_1	4	18	18
τ_2	23	25	25
τ_3	5	24	24

Answer the following, and *clearly* motivate your answers.

1. Is $\mathcal{T}$ schedulable by Global EDF?
2. Is $\mathcal{T}$ schedulable by Partitioned EDF?
3. Is $\mathcal{T}$ feasible?

Upload your answers to the above questions *as a single file*. It is recommended that you draw and write on paper and scan/photograph the page before uploading. Alternatively, you may draw the schedule and write your answer in the software of your choice. Supported file types for the upload are .jpg, .png and .pdf.

Upload your file here. Maximum one file.

The following file types are allowed: **.jpg,.png,.pdf** Maximum file size is **1 GB**

Select file to upload

A maximum of 10 points can be given for this question.

Maximum marks: 10

10 In the Mixed-Criticality sporadic task model each task τ_i is represented by five parameters. The relative deadline D_i and minimum inter-arrival separation (or period) T_i are defined as for ordinary sporadic tasks. However, each mixed-criticality task has two estimates for its worst-case execution time: C_i^{LO} and C_i^{HI}. Each mixed-criticality task also has a parameter $\chi_i \in \{LO, HI\}$ that specifies the criticality level of the task τ_i.

Upload answers to all of the below.

1. Clearly explain the meaning of the three new task parameters C_i^{LO}, C_i^{HI} and χ_i.
2. Clearly explain the correctness criteria of a set of mixed-criticality tasks. That is, explain under which circumstance certain jobs need to meet their deadlines in order for the scheduling to be considered correct.
3. Show how the recurrence relation for worst-case response-time R_i for ordinary sporadic tasks can be improved for use with mixed-criticality sporadic tasks that are scheduled using Fixed-Priority scheduling on one preemptive processor. Write down the updated recurrence relation and explain clearly why it could be used to test correctness according to the correctness criteria that you gave for point 2 above.

Upload your answers to the above questions *as a single file*. It is recommended that you draw and write on paper and scan/photograph the page before uploading. Alternatively, you may draw the schedule and write your answer in the software of your choice. Supported file types for the upload are .jpg, .png and .pdf.

Upload your file here. Maximum one file.

The following file types are allowed: **.jpg,.png,.pdf** Maximum file size is **1 GB**

Select file to upload

A total of 12 points can be given for this question.

Maximum marks: 12