EXAM BOOKLET

Computer & Software Engineering

EXAM HOSTED BY:

EVIDENT

JANUARY 5TH 2026
ENGLISH VERSION

Introduction

Welcome to the Software and Computer Engineering event of the Engineering Games 2026 🥳

During this exam, you'll be plunged into a world where dreams come true thanks to your technical skills and team spirit. Together, you form a 7-person consulting team working for the fictitious company ***Nocturna Solutions***.

Your mission: to respond to the mandates entrusted to you by various customers. Each client represents a specific area of software and IT engineering—from software architecture to security, databases, UX, AI, algorithms and more! For each of these mandates, your answers will need to demonstrate both your theoretical understanding and your ability to apply your knowledge.

The exam lasts a total of 3h30. You are encouraged to manage your time strategically to cover as many mandates as possible.

In addition to this paper leaflet (which contains the context and spaces for answering the questions), you have access to a **GitHub repo**. This repo contains all the questions that can be answered in code. Instructions for handing in your work should have been made clear on the day of the exam.

Good luck to everyone! And remember, it's not just by dreaming that we advance technology, it's by putting our skills to work for real 😉

	Possible points	Points obtained
Mandate 1 – Architecture	13	
Mandate 2 – Database	23	
Mandate 3 – Algorithms	32	
Mandate 4 – Security	20	
Mandate 5 – Devops	12	
Mandate 6 – Backend	14	
Mandate 7 – Systems	11	
Mandate 8 – UI/UX	14	
Mandate 9 – Digital logic	13	
Mandate 10 – AI	12	
Mandate 11 – Computer Creativity	28	
Total	192	

Mandate 1 - Architecture - Rêva Construction (13 points)

Customer **Rêva Construction** wants to build software systems as solid as their physical infrastructures. They are consulting you today to help them achieve a clear, modular architecture capable of supporting their projects over the long term.

Rêva Construction wants to develop an application that lets customers design and order custom beds and mattresses. The app is a web-based platform accessible on computer and mobile, which guides users through a personalized configuration process. Customers can create a profile, fill in a detailed form about their sleeping preferences and explore different bed options, materials and accessories.

The application uses a **complex algorithm** to automatically calculate the best bed for each user, considering their specific preferences and constraints. It then generates an interactive preview of the chosen bed and enables the order to be placed directly online via an integrated purchasing interface. As for Rêva Construction, the team needs to be able to manage the models offered, track orders, and consult sales statistics to improve service.

1. **Modular architecture (3 points)**
   Explain the key principles of a *cloud software architecture* and justify why they are essential to Rêva Construction's needs.

2. **Technology stack (3 points)**
   Propose a complete technological stack to develop this application, including the choice of frontend, backend and database, as well as any tools or libraries needed to manage the business logic and complex algorithm. Justify your choices in terms of application requirements and overall product quality.

3. **Software architecture (4 points)**
   Propose a software architecture suited to Rêva Construction (e.g. microservices, monolithic, layered architecture, etc.), detail its components and explain your choices. Provide a deployment diagram for your solution.

4. **Quality and sustainability (3 points)**
   Identify 4 tools, practices or methodologies to guarantee code quality, maintainability and long-term system sustainability. Justify your answers.

Mandate 2 – Database - Somnia Data (23 points)

Somnia Data is a company specializing in data management and analysis, with the slogan: "We keep your dreams in mind". It aims to develop a web-based platform enabling users to store their sleep profiles, receive personalized recommendations and follow intervention plans designed to improve the quality of their sleep.

Each user must have an account on the platform before being able to access the services. A user is characterized by a unique identifier, a first name, a last name, an address (street, city, postal code) and an e-mail address. Users can save multiple sleep profiles, each including detailed preferences (mattress type, firmness, temperature, sleeping position, night-time habits).

For each profile, the platform can generate several personalized intervention plans, identified by a unique number and a creation date. Each intervention plan can contain several recommendations (relaxation exercises, habit adjustments, sleep schedules), a description and a status indicating whether it is active, completed or pending.

For administration and performance optimization, the platform keeps track of logs and statistics on profiles, action plans and recommendations. Each log is identified by a unique identifier and contains the action taken, a description and the date. Logs are rarely accessed and are used for debugging purposes only.

Your task is to model a database designed to manage the Somnia Data platform. You'll need to identify all entities, their attributes, associations between them, cardinalities and association types (binary 1:n, n:m, etc.).

** Some questions require you to rely on your answers to questions 1 and 2. If your answers differ from those in the solution, we'll correct them according to your answers. If you feel unable to answer the first questions, we can provide you with the answers. This results in zero points for the first 2 questions*

1. **Entity-Association diagram (6 points)**
   Draw a UML Entity-Association diagram or relational schema representing the database required for the Somnia Data application. Don't worry about the particular conventions of the representation language you've learned. It is important that you identify all the entities, their keys, attributes, associations and type (binary, 1:n, n:m, etc.).

2. **SQL schema (4 points)**
   Using your Entity-Association Diagram, write the SQL schema in the file *SCHEMA.sql* of Mandate 2 - Somnia Data in the Github repo. Use MySQL if you can.

3. **SQL queries (5 points)**
   For the schema you proposed in question 2, write the SQL queries in the corresponding *REQUEST_X.sql* files in Mandate 2 - Somnia Data in the Github repo. Use MySQL if you can. Here's what to look for in your queries.

   a. Retrieve the list of users with more than 3 active intervention plans.

   b. Retrieve a list of all the profiles of a given user (user id = 42) with its number of pending intervention plans.

   c. Retrieve all non-active recommendations for a given plan (plan id = 30).

   d. Retrieve the history of actions in the logs for a given user (user id = 42) on his plans and recommendations, sorted in descending order according to their creation date.

   e. Retrieve all intervention plans with their associated profile and user created between December 1st and January 1st, sorted chronologically by creation date.

4. **Indexing and performance (3 points)**
   For the schema you proposed in question 2, indicate where you will add indexes to improve query performance. Justify each index: why is it necessary, on which column(s) and for which type of query. Also explain the general advantages and disadvantages of using indexes in SQL.

5. **Object-oriented database (2 points)**
   Explain **the main differences** between a relational database (SQL) and an object/document-oriented database (NoSQL).

6. **Final thoughts (3 points)**
   In your opinion, which type of database (SQL) or (NoSQL) is best applied to Somnia Data's needs? Justify your answer with the strengths/weaknesses you identified in question 5.

Mandate 3 – Algorithms - Algoria Labs (32 points)

Algoria Labs is a research company specializing in the design of algorithms capable of transforming dreams into concrete solutions. Their playground: optimizing, analyzing and solving complex problems where logic and creativity meet.

Your team is assigned a series of algorithmic challenges inspired by the world of sleep and dreams.

1. **Dream Navigation (6 points)**
   Algoria Labs' *Dream Navigation™* department studies travel mechanisms in lucid dreams. Their research focuses on optimal navigation in dream environments where traditional physical laws don't always apply. Your objective is to develop an intelligent path-finding algorithm for navigating complex dream labyrinths containing teleportation portals, acceleration zones and temporal obstacles. Code and instructions are available on the Github repo.

2. **To Infinity and Beyond (6 points)**
   Algoria Labs' *To Infinity And Beyond™* department wants to stack blocks of different dimensions to form towers as high as possible, with the ambition of reaching for the stars. Your objective is to develop an intelligent optimization algorithm to maximize the stacking height of blocks while respecting the constraints of strength, dimensions and fragility. The code and instructions are available on the Github repo.

3. **Tic Tac Throw! (7 points)**
   Algoria Labs' *Idiot Factory™* department is interested in a very particular problem: developing "intelligent" agents... that always lose. Your objective will be to implement an artificial agent that plays a modified 4x4 version of Tic Tac Toe in a systematically sub-optimal way, in order to maximize its chances of losing against various types of opponents. Code and instructions are available on the Github repo.

4. Asymptotic complexity (3 points)

For each of the following code extracts, evaluate the worst-case asymptotic complexity. Indicate, for each function, the time complexity and the space complexity, using large O notation.

```python
def func1(n):
    total = 0
    i = 1
    while i < n:
        j = i
        while j > 0:
            total += 1
            j //= 2
        i *= 2
    return total
```

Time complexity:

Spatial complexity:

```python
def func2(n):
    if n <= 1:
        return 1
    return func2(n - 1) + func2(n // 2) + func2(1)
```

Time complexity:

Spatial complexity:

```python
def func3(n):
    arr = [0] * n
    for i in range(n):
        j = i
        while j < n:
            arr[j] += 1
            j += (j - i + 1)
```

Time complexity:

Spatial complexity:

```python
def func4(n):
    if n <= 0:
        return 0
    if n == 1:
        return 1
    return func4(n - 1) + func4(n - 1)
```

Time complexity:

Spatial complexity:

```python
def func5(n):
    s = set()
    for i in range(n):
        for j in range(i):
            s.add((i, j))
        for k in s:
            x = k[0] * k[1]
```

Time complexity:

Spatial complexity:

5. Applied data structures (5 points)

For each of the following scenarios, determine which data structure (Heap, Stack, Tree, Linked List, etc.) is best applied to the problem. Justify your answer in terms of the spatial and temporal complexity of the operations of the chosen structure, and how this applies to the needs of the problem.

a. We want to manage a real-time stream of integers representing active session IDs on a server.
- At each stage, we need to insert a new ID, check whether an ID is active, and remove expired IDs.
- Operations must be fast.

b. We want to analyze a massive stream of incoming numbers.
- We want to be able to keep the K largest elements seen so far,
- We want to be able to retrieve the minimum of these K at any time
- We want to update the structure efficiently when a new number arrives.

c. We want to code an autocomplete feature in a messaging app.
- Starting with a prefix, you need to return all stored words that begin with that prefix.
- The dictionary contains millions of words.
- The queries must be fast and the memory well optimized.

d. We want to manage a sequence of numbers that can change over time.
   - We need to be able to query the minimum in a range [l, r] several times.
   - Queries and updates must be fast.

e. We want to develop a routing system for a transport network.
   - The graph is sparse, weighted, with non-negative weights.
   - We need to be able to find the shortest path between two cities several times a second.

6. Sets and relationships (2 points)

Let the relation R be defined on the set {1,2,3,4} by :
R = {(1,1), (2,2), (3,3), (4,4), (1,2), (2,3), (3,4)}

Determine which properties the relation R possesses:

a) Reflexive, symmetrical, transitive, and a partial order relationship

b) Reflexive and transitive, but not symmetrical; a partial order relationship

c) Symmetrical and transitive, but not reflexive; not a partial order relationship

d) Reflexive only, neither symmetrical nor transitive; not a partial order relationship

7. Graphs (2 points)

Let the graph G be defined by the vertices V = {A, B, C, D, E} and edges E = {{A,B}, {A,C}, {B,C}, {B,D}, {C,E}}.

Which of the following statements correctly describes the properties of this graph?

a) The graph is connected, contains a cycle and is not bipartite

b) The graph is connected, contains no cycles, and is bipartite

c) The graph is not connected, contains a cycle, and is bipartite

d) The graph is not connected, contains no cycles and is not bipartite

8. Trees (1 point)

Consider a complete binary tree of height h = 4.
Which statement correctly describes the maximum number of vertices (or nodes) and leaves in this tree?

a) Maximum of 15 vertices and 7 leaves

b) Maximum of 31 vertices and 16 leaves

c) Maximum of 16 vertices and 8 leaves

d) Maximum of 30 vertices and 15 leaves

Mandate 4 - Security - Nightmare Reaper (20 points)

Nightmare Reaper is a cybersecurity company specializing in Red Team testing. Its aim is to identify and exploit various vulnerabilities in order to demonstrate the associated risks to its customers and partners, and to demonstrate its methodology.

1. **Capture the Flag (3x3 points)**
   Your objective is to identify several hidden *flags* (in ***FLAGXYZ*** format), each representing a distinct vulnerability. The code to be investigated and the instructions can be found in the Github repo.

   For each flag, describe in a structured and reproducible way: the chronological steps and precise actions; the tools, parameters and techniques used (e.g. scan, brute-force, injection, escalation); and justify your answers.

2. **Steganography (3x3 points)**
   Your objective is to identify several hidden *flags* (in ***FLAGXYZ*** format), each representing a way of hiding information in images, text etc. The files and instructions can be found on the Github repo.

   For each flag, describe in a structured and reproducible way: the chronological steps and precise actions; the tools, parameters and techniques used (e.g. scan, brute-force, injection, escalation); and justify your answers.

3. **Password (2 points)**
A password is a sequence of 5 characters chosen from the 26 uppercase letters of the English alphabet and the 10 numbers (0-9), i.e. 36 possible characters.

Select the statement that is true about the number of possible passwords in the following 3 situations:

- Unrestricted
- At least 2 distinct characters
- All distinct characters

a) - Unrestricted: 60,466,176
- 2 separate letters: 6500
- All distinct characters: 45239040

b) - Unrestricted: 60,466,176
- At least 2 distinct characters: 60 466 140
- All distinct characters: 45239040

c) - Unrestricted: 60,466,176
- At least 2 distinct characters: 1,413,720;
- All distinct characters: 60,466,176

d) - Unrestricted: 67,523,422
- At least 2 distinct characters: 10×C(5,2)
- All distinct characters: 60,466,176

Mandate 5 – Devops - Oléoducs Scripts (12 points)

Oléoducs Scripts is an operations development company with expertise in Docker technologies and scheduling.

You'll be responsible for patching / setting up a Docker environment that manages its networking with various applications running in containers.

1. **Docker config (5 points)**
   Run the environment with "hardcode" values (fixed url, fixed port, fixed number of containers, etc.). The code to complete and the instructions can be found in the Github repo.

2. **Scale up (5 points)**
   Added "Scale up" functionality with associated scripts (number of "slaves" that can be changed with a single parameter and that can be contacted by the load-balancer. The code and instructions are available on the Github repo.

3. **Blue green deployment (2 points)**
   Consider a Kubernetes cluster running a stateless microservice. A developer has implemented a progressive update using the following configuration:

```yaml
spec:
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 0
```

   Which of the following situations would not be a valid use of this configuration? Circle your answer and justify why.

   a) Progressive upgrading of a service with a large number of users while maintaining full availability.

   b) Test a new feature on a subset of instances before rolling it out to all instances.

   c) Ensure no downtime during critical business hours.

   d) Remplace a faulty version of the service with a corrected version

Mandate 6 - Backend - Evening Dream (14 points)

Evening Dream is a young software development company specializing in the art of partying (organizing drunken evenings with friends). Your team is tasked with designing and implementing a backend server capable of supporting an application that will help users plan their evening.

1. **Backend server (8 points)**
   Write a server that can write the general planning of a night out. The server will need to be able to communicate with various external APIs to get information about how to build this party. The code to be completed and the instructions can be found in the Github repo.

2. **Cache (3 points)**
   In creating a cache for your service, specifically on the input that requires a lot of processing (see the route that creates the "evenings") you expect to receive up to 20 parameters that can be used at the same time (these parameters have an average cardinality of 100). How would you proceed to cache part of the data to reduce server load? *Considerations*: The process would follow that for the previous question, but with a greater possibility of parameters, and it's a*key-value* style

3. Ticket to Ride (3 points)

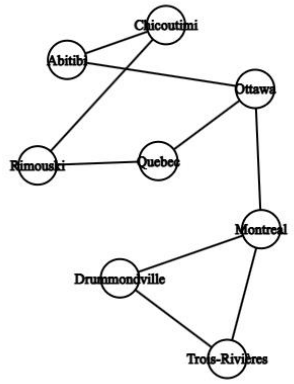
For the application developed for Evening Dream, it will be necessary to design specialized algorithms to validate new *Ticket to Ride* game boards.

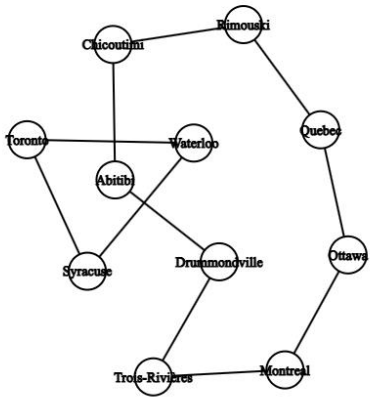
Your task will be to describe some of the maps created by the company using concepts derived from graph theory, and then to associate the relevant terms from the word bank with each map.

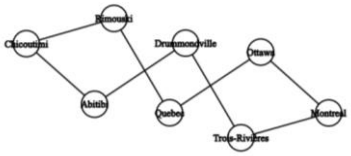
Example: A map "X" contains both Hamiltonian and Eulerian paths but is not connected.

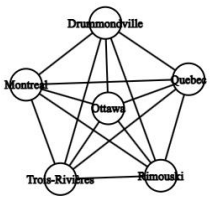
Bank of terms to apply:

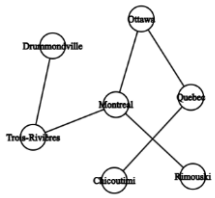
Hamiltonian	Eulerian	Connected (/unconnected)
Planar	Biconnected	Complete
Regular	Acyclic	











Mandate 7 - Systems - DreamKernel (11 points)

DreamKernel is a low-level IT development company that takes care of your low-level operations and/or application needs.

1. **Deadlocks (5pts)**
   One of their applications attempts to build random strings like Cloudflare and its famous wall of "Lava lamps", but the initial stream, written by a recently departed employee, lacks tolerance to deadlocks. Indeed, the application should be able to run forever, but it ends up in a state of continuous waiting, with no way of getting out of it. Your aim is to make the application resilient to this problem. The code to complete and the instructions can be found on the Github repo.

2. **Locks (3 points)**
   Many systems in common use today use a mix of Mutex and Semaphore for reading/writing data. Explain the difference between a Mutex, a Semaphore, and how a DB would use these concepts for its reads / writes (you can ignore the transaction concept that SQL and other technologies offer).

3. Threads (3 points)

According to the following code, determine the possible outputs of this program through its standard output. Explain the process that makes these outputs possible, and then explain the difference it would make if a sleep(1) were applied between each pair of printf instructions (A1-A2, B1-B2, C1-C2).

```c
#include <stdio.h>
#include <pthread.h>

void* workerA(void* arg) {
    printf("A1\n");
    printf("A2\n");
    return NULL;
}

void* workerB(void* arg) {
    printf("B1\n");
    printf("B2\n");
    return NULL;
}

void* workerC(void* arg) {
    printf("C1\n");
    printf("C2\n");
    return NULL;
}

int main() {
    int thread = 0;
    pthread_t t1, t2, t3;

    if (thread == 0) {
        pthread_create(&t1, NULL, workerA, NULL);
        pthread_create(&t2, NULL, workerB, NULL);
        pthread_create(&t3, NULL, workerC, NULL);

        pthread_join(t1, NULL);
        pthread_join(t2, NULL);
        pthread_join(t3, NULL);

        printf("Main done\n");
    }
    return 0;
}
```

Mandate 8 - UI/UX - Evident – Partner Question (14 points)

Evident is an international company recognized for its non-destructive testing solutions that detect internal defects such as cracks, porosities, voids, or irregularities in various materials. The company also develops software solutions specifically designed for technicians and inspectors in the field, who are generally non-IT specialists. These tools must be extremely easy to use, robust, ergonomic, and operate reliably in demanding environments (factories, worksites, noisy or dimly lit areas).

As part of this fictional scenario, Evident wants to explore new UX approaches for future digital tools. An experimental (fictional) prototype has therefore been developed: a small web application that allows users to record inspection data, view simple metrics, and receive automatic recommendations.

However, despite functioning correctly from a technical standpoint, this fictional prototype has generated several negative comments from test users. Evident would therefore like a critical analysis of the user experience.

Your task: perform a UX critical evaluation of the software by identifying potential obstacles and proposing improvements. The (fictional) prototype can be found in the GitHub repository of Mandate 9, where you will find the application code (HTML/JS/CSS) as well as instructions to run it.

1. **Cognitive load (2 points)**
   Analyze the protype's interface and identify two elements that could increase users' cognitive load. Propose a solution to reduce this load while maintaining functionality.

2. **Dark patterns (2 points)**
   Explain what a UI/UX dark pattern is. Identify a dark pattern in the application, Justify why it's a dark pattern and how you would fix it.

3. **UX gaps and improvements (4 points)**
   - Identify at least 3 gaps or friction points in the tool's user experience.
   - For each gap, make a link to Tognazzini's usability principles (link in the repo) and how these are not followed.
   - Then propose a concrete improvement for each gap, justifying how it would solve the problem.

4. **Modern interface (6 points)**
   As you have probably noticed, the current interface does not meet modern standards of ergonomics and readability. Your mission: modify the files *index.html* and *styles.css* while keeping all functional elements intact. Your objective is to make the application more "clean" and modern, but adapted to the appropriate type of user. The evaluation will be done in comparison with the other teams. The code to complete, as well as the instructions, can be found in the GitHub repository.

Mandate 9 - Digital logic - Wired Dreams (13 points)

Wired Dreams, popular for low-level, rambling designs, is asking you to help them create their LED power supply logic circuits with unusual inputs and sequences.

1. **Designing equations (5 points)**
   Make the Karnaugh table and give the simplified equation (minterm or maxterm) for opening an LED, according to the following instructions. Entries are labelled A, B, C, D, E and if two setpoints conflict, priority goes to the first setpoint, followed by the second etc. Cases ignored by the setpoints are considered "DON'T CARE".

   Consideration: Equations can use exclamation marks, apostrophes or a line above the whole formula, or ¬ for **negation**, ^**(and**), v**(or**), etc.
   However, be careful to use a valid form of notation, otherwise penalties may apply. For example, "**!A**'" is not valid.
   * Empty Karnaugh tables are available to help you

   1e. LED is **ON** if (A and B), but not (C or !E).
   2e. The LED is **ON** if D is lit.
   3e. LED is **CLOSED** if B or E is lit.
   4e. LED is **ON** if (C and E), but not B.
   5e. LED is **ON** if !A and (B or C).
   6e. LED is **CLOSED** if A and D are ON.
   7e. The LED is **ON** if exactly two of inputs A, C, E are set to 1.
   8e. The LED is **ON** if all inputs are at 0.
   9e. LED is **CLOSED** if C is ON and E is OFF.

Example of rule priorities
If I have the rules

1. If A or B => LED ON (priority)
2. If A and B => LED Closed

A \ B	0	1
0	X	1
1	1	1

But if I have the rules

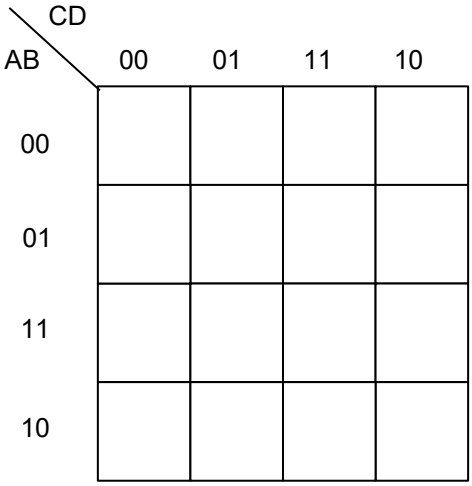
1. If A and B => LED Closed (priority)
2. Si A or B => LED Allumé

A \ B	0	1
0	X	1
1	1	0

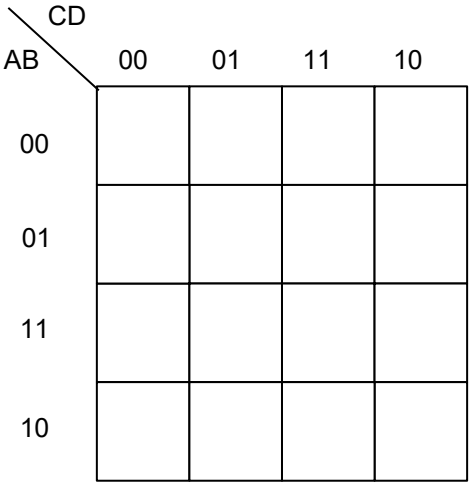
Empty karnaugh tables

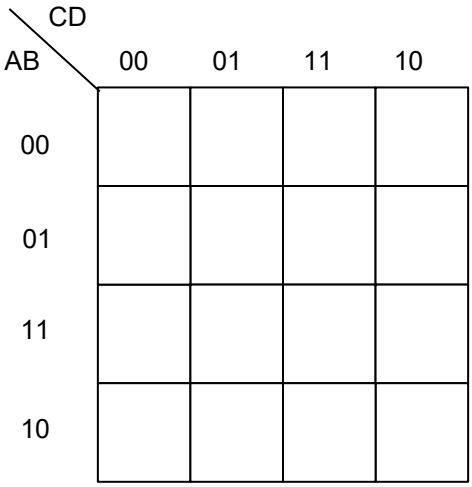
(Cross out draft tables if you use them)

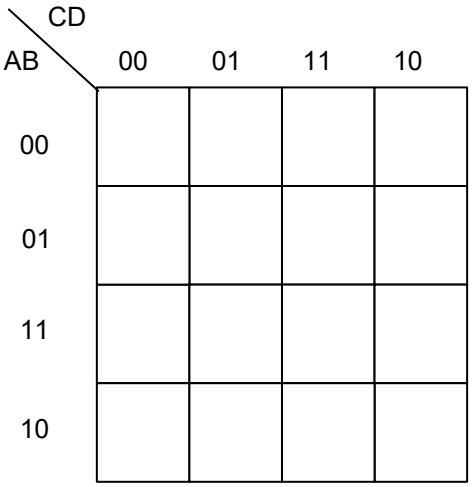
E=0



E=1







Enter your formula here:

2. Circuit and Mealy (5 points)

Design the electrical circuit and its mealy machine(s), using only D-Latch, AND, OR and inverters, for the following situation.

Consider 4 buttons: **B1, B2, B3 and R**
And 2 LEDs: **L1 and L2**

Rules for **R:**

This is the **reset** button: whatever the state, it returns to the initial state.

Rules for **L1**:

If I press ***B1***, followed by ***B3*** followed by ***B2*** and ***B3*** at the same time, ***L1*** is lit, and remains *lit* until ***R or B1*** followed by ***B2*** is *pressed*.

Rules for **L2**:

If I press ***B1*** twice in a row, I won't need to press ***2xB1*** again to *switch on* ***L2***; any wrong move afterwards reverts to the last completed ***2xB1***. The only thing that skips the state is ***R***.
The *activation* sequence is as follows: ***2xB1***, ***1xB2, 2xB1***, ***B3***.
To deactivate **L2** once *lit*: **R or B1x2**

3. Proofs (2 points)

Consider the following proposition: P(n): n^2-n is even for any integer n≥0. We wish to demonstrate this proposition by induction. What sequence of reasoning can be used to correctly validate this demonstration?

a) - P(0) is false
- The induction hypothesis assumes that k^2 - k is even
- *(k+1)^2 - (k+1) = k^2 - k + 2k*
- We deduce that P(n) is true for even n only.

b) - P(0), P(1) and P(2) are true
- The induction hypothesis assumes that *k^2 - k* is even
- *(k+1)^2 - (k+1) = k^2 - k + 2k + 2*
- Since the sum of two even numbers is even, P(k+1) is true
- We therefore deduce that P(n) is true for all n >= 0.

c) - P(0), P(1) and P(2) are true
- P(0) is true, but P(1) is false
- The induction hypothesis assumes that *k^2 - k* is odd
- *(k+1)^2 - (k+1) = k^2 - k + 2k + 2*
- We therefore deduce that P(n) is true for odd n only.

d) - Base verification is sufficient
- There's no need to use induction
- We therefore deduce that P(n) is true for all even n.

4. Predicates (1 point)

Let the formula be: (p∨q)∧(¬p∨r)
Select the nature(s) of this formula:

a) Tautology

b) Contradiction

c) Satisfactory

Mandate 10 – AI - SchleepShop (12 points)

SchleepShop is a company specializing in the development of artificial intelligence systems to improve sleep quality. Their aim is to offer personalized recommendations based on users' sleep habits, physiological data and preferences.

The SchleepShop platform uses machine learning algorithms to :

- Predict the best sleep conditions for a given user.
- Generate personalized sleep plans.
- Detect anomalies or trends in sleep behavior.

The company wants your team to evaluate and propose theoretical solutions to improve their models and the use of AI in the platform.

1. **Predictive models (3 points)**
   Explain what supervised learning methods could be used to predict a user's sleep quality based on their habits. Consider that SchleepShop has a huge amount of data on its users' habits and sleep metrics. Justify why you think these methods would be best applied in this context, presenting the advantages and limitations of each proposed method.

2. **Data pre-processing (2 points)**
   Identify the types of data relevant to training sleep prediction models (Be creative). Based on this data, explain which pre-processing steps are necessary to obtain reliable, usable data.

3. **Evaluation and validation (2 points)**
   Propose an evaluation method to measure the performance of predictive models.
   Explain how to avoid overfitting and ensure model generalization.

4. **Ethics and bias (2 points)**
   Identify 3 risks of bias in personalized sleep recommendations. Propose a strategy to minimize each bias and ensure ethical use of AI.

5. **Prompt Engineering (3 points)**
   SchleepShop wants to use generative AI to create their personalized sleep plans. Help them write robust prompts to enable their AI to return a message they can use later. You're not allowed to test your prompts since the AI is restricted, so use only prompt engineering theory to design your prompts. Code to complete and instructions can be found in the Github repo.

Mandate 11 - Computer Creativity - Nocturna Solution (28 points)

Nocturna Solutions, your own company **Nocturna Solutions** also needs to evolve and improve its internal practices. As a team of 7 consultants, you are sometimes called upon to work on more creative, playful or organizational projects that directly concern the company's culture and image.

1. **Logology (7 points)**
   Nocturna Solutions wants to modernize its image and create an original logo. However, due to a lack of budget, the marketing team decided that the logo should be generated in ASCII art directly from code. The code to complete and instructions can be found on the Github repo.

2. **Sloganology (7 points)**
   In line with its new logo, Nocturna aims to provide an intelligent slogan generation tool. The quality of the solution will be assessed according to originality, the quality of the results in relation to the company, the uniqueness of the results and their grammatical correctness. The code and instructions can be found on the Github repo.

3. **Officology (7 points)**
   Nocturna Solutions wants to redefine its ideal office to maximize the enjoyment and productivity of its consultants. The workspace is represented as a 2D grid of available tiles. Several elements need to be strategically placed in this space, such as: consultants' desks, plants, a coffee machine, nap chairs and others. Your objective is to propose an optimal layout to maximize the team's total joy value. Code to complete and instructions can be found in the Github repo.

4. **Emojiology (7 points)**
   Nocturna Solutions wants to analyze its employees' sleep patterns. To do this, it has data on the start time of sleep, and the state of sleep at each subsequent hour for its employees. The aim is to determine the average sleep pattern of its employees at each hour of the day. The twist? The data inputs are in emojis, the code must be written in *Emojicode* and the output must be provided in emojis too. The code to be completed and the instructions can be found in the Github repo.

End of exam 😎