Problem Description

Alex, an undergraduate student at SUTD currently in Term 4 of the Engineering Systems and Design (ESD) programme, aims to develop a weekly timetable to improve his time management. The schedule will be tailored to accommodate his academic modules, meals, breaks, leisure activities, and fifth row commitments.

Decision Variables

A weekly timetable is constructed using one-hour time slots, with each day comprising 15 slots from 8:00 AM to 11:00 PM. The remaining hours, from 11:00 PM to 8:00 AM, are allocated for Alex to wash up, sleep, and eat his breakfast.

- $x_{i,j,k} \in \{0, 1\}$
  Binary variable indicating whether activity k is scheduled on day i at time slot j.
  - $x_{i,j,k} = 1$ if activity k is scheduled at time slot j on day i
  - $x_{i,j,k} = 0$ otherwise

  Where:
  - $i \in \{1, 2, \ldots, 7\}$ represents the days of the week (Monday to Sunday)
  - $j \in \{8, 9, \ldots, 22\}$ represents one-hour time slots from 8am to 11pm
  - $k \in \{1, 2, \ldots, 9\}$ represents the activities:
    * $k = 1$: Optimisation
    * $k = 2$: Probability & Statistics
    * $k = 3$: Data & Business Analytics
    * $k = 4$: HASS
    * $k = 5$: Lunch
    * $k = 6$: Dinner
    * $k = 7$: Exercise
    * $k = 8$: Leisure
    * $k = 9$: Fifth Row

Constraints

1. Time Slot Exclusivity Constraint

To ensure that Alex is not scheduled for more than one activity at any given time slot, we impose the following constraint:

$$\sum_{k=1}^{9} x_{i,j,k} \leq 1 \quad \forall i \in \{1,\dots,7\}, \quad \forall j \in \{8,\dots,22\}$$

- This constraint ensures that only one activity can be scheduled per time slot on any given day.

2. Daily Limit on Study Hours for Each Subject

$$\sum_{j=8}^{22} x_{i,j,k} \leq 2 \quad \forall i \in \{1,\dots,7\}, \quad \forall k \in \{1,2,3,4\}$$

- This constraint ensures that each subject (activity 1 to 4) can be scheduled for **at most 2 hours per day** (day i). This prevents overloading on a single subject in one day.

3. Minimum Duration of Study and Leisure Session

To promote effective learning and minimize the inefficiencies associated with frequent task-switching, each study session must last at least **2 consecutive hours**, and leisure should consist of exactly **1 session of 5 consecutive hours during the weekend**. We define an indicator variable:

- $s_{i,j,k} \in \{0,1\}$
  Indicates whether activity $k \in \{1,2,3,4,8\}$ (study or leisure) starts at time slot j on day i.

This is enforced using the following constraints:

Start detection constraints:

$$x_{i,j,k} - x_{i,j-1,k} \geq 1 - 2(1 - s_{i,j,k}) \quad \forall k \in \{1,2,3,4,8\}$$

$$x_{i,j,k} - x_{i,j-1,k} \leq s_{i,j,k} \quad \forall k \in \{1,2,3,4,8\}$$

Minimum duration constraints for subjects:

- For study subjects $k \in \{1,2,3,4\}$ (minimum 2-hour blocks):

$$\sum_{\tau=j}^{j+1} x_{i,\tau,k} \geq 2 \cdot s_{i,j,k} \quad \forall i, \forall j \in \{8,\dots,21\}, \forall k \in \{1,2,3,4\}$$

- Dealing with an edge case (j=22), which was not defined in the previous constraint. To ensure that no new study session is started at the final hour of the day (hour 22), since a valid 2-hour session cannot begin there:

$$s_{i,22,k} = 0 \quad \forall i, \ \forall k \in \{1,2,3,4\}$$

- Another edge case: This constraint prevents isolated 1-hour sessions at hour 8 by requiring that any study activity at hour 8 must be followed by activity at hour 9.

$$x_{i,8,k} \leq x_{i,9,k} \quad \forall i, \ \forall k \in \{1,2,3,4\}$$

Unique Leisure Session Constraint (Weekend only): Alex wants a 5h block of time in the weekend to hang out with his friends or partake in leisure activities.

$$\sum_{i=6}^{7}\sum_{j=8}^{18} s_{i,j,8} = 1$$

$$\sum_{i=1}^{7}\sum_{j=8}^{22} x_{i,j,8} = 5$$

- The first constraint ensures exactly one 5-hour leisure block starts on either Saturday or Sunday. The second cosntraint disallows any other time slots to be taken up by leisure.
- The time index $j \in \{8, \ldots, 19\}$ ensures enough time remains in the day for a 5-hour block.

Correctness of the $s_{i,j,k}$ Indicator Variable

The binary variable $s_{i,j,k}$ correctly detects the start of a study or leisure session. When $s_{i,j,k} = 1$, it signifies that activity k starts at hour j on day i, and this activates constraints enforcing a minimum duration (e.g., two consecutive hours for study sessions).

- If $s_{i,j,k} = 1$, then $x_{i,j,k}$ and $x_{i,j+1,k}$ must both be 1 to ensure continuity, satisfying the minimum session length.
- If $s_{i,j,k} = 0$, the start constraint is not triggered, allowing continuation from a previous time slot (i.e., $x_{i,j,k} = 1$ is valid if the session was already ongoing).
- The formulation handles edge cases:
  - At $j = 22$, $s_{i,22,k}$ is forced to 0 to prevent starting a session that cannot meet the 2-hour requirement.
  - At $j = 8$, constraints ensure that if $x_{i,8,k} = 1$, it must be followed by $x_{i,9,k} = 1$, avoiding isolated one-hour sessions.

4. Weekly Study Hour Requirements (Per Module)

To ensure Alex spends enough time on each module throughout the week (including both pre-allocated lessons and self-study), we impose the following constraint for each academic activity $k \in \{1, 2, 3, 4\}$:

$$\sum_{i=1}^{7}\sum_{j=8}^{22} x_{i,j,k} \geq R_k \quad \forall k \in \{1, 2, 3, 4\}$$

Where R_k is defined as:

- $R_1 = 12$ hours (Optimisation)
- $R_2 = 12$ hours (Probability & Statistics)
- $R_3 = 10$ hours (Data & Business Analytics)
- $R_4 = 8$ hours (HASS)

These values are based on course guideline recommendations (self study should be 1.5x lesson time), which suggest a suitable weekly study duration to support adequate understanding and academic performance in each subject.

5. Lunch Constraint

To ensure that Alex has lunch every day between 11am and 2pm, we impose the following constraint:

$$\sum_{j=11}^{13} x_{i,j,5} = 1 \quad \forall i \in \{1, \ldots, 7\}$$

$$\sum_{j=8}^{22} x_{i,j,5} = 1 \quad \forall i \in \{1, \ldots, 7\}$$

- This ensures exactly one time slot per day is allocated to lunch (activity $k = 5$) within the 11am–2pm window. The second constraint is necessary to prevent lunch from appearing in other time slots not covered by the first constraint.

6. Dinner Constraint

To ensure that Alex has dinner every day between 5pm and 8pm, we impose the following constraint:

$$\sum_{j=17}^{19} x_{i,j,6} = 1 \quad \forall i \in \{1, \ldots, 7\}$$

$$\sum_{j=8}^{22} x_{i,j,6} = 1 \quad \forall i \in \{1, \ldots, 7\}$$

- This ensures that one time slot per day is allocated to dinner (activity $k = 6$) within the 5pm–8pm window. Similarly, the second constraint is necessary to prevent dinner from appearing in other time slots not covered by the first constraint.

7. Exercise Constraint (3 Sessions per Week, No Consecutive Days)

Alex intends to exercise thrice a week, one hour each time. He wants to avoid exercising on back-to-back days. Exercise is activity k = 7.

- **Exactly 3 exercise sessions per week:**

$$\sum_{i=1}^{7} \sum_{j=8}^{22} x_{i,j,7} = 3$$

- **No exercise on consecutive days (including Sunday and Monday):**

Let $e_i \in \{0,1\}$ be a binary variable that is 1 if exercise is scheduled on day i.

$$e_i \geq \sum_{j=8}^{22} x_{i,j,7} \quad \forall i \in \{1,\ldots,7\}$$

Enforce the non-consecutive condition:

$$e_i + e_{i+1} \leq 1 \quad \forall i \in \{1,\ldots,6\}$$

$$e_7 + e_1 \leq 1$$

Correctness of the e_i Indicator Variable

The binary variable e_i is used to detect whether exercise occurs on day i. Its correctness is enforced by the following formulation:

- If any exercise session is scheduled on day i, then $\sum_{j=8}^{22} x_{i,j,7} \geq 1$, which forces $e_i = 1$ due to the constraint $e_i \geq \sum_{j=8}^{22} x_{i,j,7}$.
- If no exercise is scheduled on day i, then the right-hand side of the inequality is 0, allowing $e_i = 0$.
- The indicator e_i is then used to impose the non-consecutiveness constraint:

  $$e_i + e_{i+1} \leq 1 \quad \forall i \in \{1,\ldots,6\}, \quad e_7 + e_1 \leq 1$$

  which ensures that no two exercise days occur back-to-back, even across the Sunday–Monday boundary.

8. Preallocated Timetable (Fixed Lessons and Fifth Rows)

These are fixed and cannot be adjusted during optimization. They are modeled as hard constraints in the model.

$$x_{i,j,k} = 1 \quad \text{for all preassigned lessons and 5th row activities}$$

$$\begin{aligned}
&x_{1,10,1} = 1, \quad x_{1,11,1} = 1, \quad x_{1,15,4} = 1, \quad x_{1,16,4} = 1,\\
&x_{2,10,2} = 1, \quad x_{2,11,2} = 1, \quad x_{2,13,1} = 1, \quad x_{2,14,1} = 1,\\
&x_{2,16,3} = 1, \quad x_{2,17,3} = 1,\\
&x_{3,19,9} = 1, \quad x_{3,20,9} = 1, \quad x_{3,21,9} = 1,\\
&x_{4,9,4} = 1, \quad x_{4,11,2} = 1, \quad x_{4,12,2} = 1,\\
&x_{5,9,3} = 1, \quad x_{5,10,3} = 1
\end{aligned}$$

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
08:00–09:00							
09:00–10:00				HASS ($x_{4,9,4}$)	DBA ($x_{5,9,3}$)		
10:00–11:00	OPT ($x_{1,10,1}$)	PNS ($x_{2,10,2}$)			DBA ($x_{5,10,3}$)		
11:00–12:00	OPT ($x_{1,11,1}$)	PNS ($x_{2,11,2}$)		PNS ($x_{4,11,2}$)			
12:00–13:00				PNS ($x_{4,12,2}$)			
13:00–14:00		OPT ($x_{2,13,1}$)					
14:00–15:00		OPT ($x_{2,14,1}$)					
15:00–16:00	HASS ($x_{1,15,4}$)						
16:00–17:00	HASS ($x_{1,16,4}$)	DBA ($x_{2,16,3}$)					
17:00–18:00		DBA ($x_{2,17,3}$)					
18:00–19:00							
19:00–20:00			5th Row ($x_{3,19,9}$)				
20:00–21:00			5th Row ($x_{3,20,9}$)				
21:00–22:00			5th Row ($x_{3,21,9}$)				
22:00–23:00							

Objective: Maximize Breaks After Any Activity (General Spacing)

Alex wants to have a balanced schedule: he would like his activities spread across the day and across the week, with adequate breaks throughout the day. These regularly spaced breaks are put, in case he needs to carry out miscellaneous activities, or for his free time.

To promote spacing and reduce back-to-back scheduling, we introduce a binary variable $\delta_{i,j} \in \{0, 1\}$, where:

- $\delta_{i,j} = 1$ if time slot $j - 1$ is any activity and time slot j is a break (i.e., no activity is scheduled); $\delta_{i,j} = 0$ otherwise.

This logic is enforced using the following constraints:

$$\sum_{k=1}^{9} x_{i,j-1,k} - \sum_{k=1}^{9} x_{i,j,k} \geq 1 - 2 \cdot (1 - \delta_{i,j})$$

$$\sum_{k=1}^{9} x_{i,j-1,k} - \sum_{k=1}^{9} x_{i,j,k} \leq \delta_{i,j}$$

Objective Function:

$$\max \sum_{i=1}^{7} \sum_{j=9}^{22} \delta_{i,j}$$

This objective maximizes the number of breaks immediately following any activity, encouraging natural spacing throughout the timetable.

Correctness of the $\delta_{i,j}$ Indicator Variable

The binary variable $\delta_{i,j}$ is used to identify if a break immediately follows an activity on the same day. It is correctly activated only when:

- Time slot $j-1$ is occupied (i.e., $\sum_k x_{i,j-1,k} = 1$), and
- Time slot j is free (i.e., $\sum_k x_{i,j,k} = 0$).

This is enforced through the following pair of constraints:

$$\sum_{k=1}^{9} x_{i,j-1,k} - \sum_{k=1}^{9} x_{i,j,k} \geq 1 - 2(1 - \delta_{i,j})$$

$$\sum_{k=1}^{9} x_{i,j-1,k} - \sum_{k=1}^{9} x_{i,j,k} \leq \delta_{i,j}$$

- When $\delta_{i,j} = 1$, these constraints reduce to:

$$\sum x_{i,j-1,k} - \sum x_{i,j,k} \geq 1 \quad \text{and} \quad \sum x_{i,j-1,k} - \sum x_{i,j,k} \leq 1$$

- This ensures exactly one activity before a break.
- When $\delta_{i,j} = 0$, the constraints become:

$$\sum x_{i,j-1,k} - \sum x_{i,j,k} \geq -1, \quad \sum x_{i,j-1,k} - \sum x_{i,j,k} \leq 0$$

- This makes $\delta_{i,j}$ inactive, regardless of the values of x.

Hence, $\delta_{i,j}$ is only 1 when a break directly follows an activity, enabling the objective to maximize such naturally spaced slots across the week.

Formulation of Integer Linear Program (ILP)

Objective:

$$\max \sum_{i=1}^{7} \sum_{j=9}^{22} \delta_{i,j}$$

Subject to:

- **Time Slot Exclusivity:**

$$\sum_{k=1}^{9} x_{i,j,k} \leq 1 \quad \forall i, j$$

- **Daily Study Limit:**

$$\sum_{j=8}^{22} x_{i,j,k} \leq 2 \quad \forall i,\ \forall k \in \{1, 2, 3, 4\}$$

- **Minimum Duration of Study/Leisure:**

$$x_{i,j,k} - x_{i,j-1,k} \geq 1 - 2(1 - s_{i,j,k}), \quad x_{i,j,k} - x_{i,j-1,k} \leq s_{i,j,k}$$

$$\sum_{\tau=j}^{j+1} x_{i,\tau,k} \geq 2s_{i,j,k} \quad \forall j \leq 21,\ \forall k \in \{1,2,3,4\}$$

$$\sum_{\tau=j}^{j+4} x_{i,\tau,8} \geq 5s_{i,j,8} \quad \forall i \in \{6,7\},\ j \leq 18$$

$$s_{i,22,k} = 0, \quad x_{i,8,k} \leq x_{i,9,k} \quad \forall i,\ \forall k \in \{1,2,3,4\}$$

$$\sum_{i=6}^{7} \sum_{j=8}^{18} s_{i,j,8} = 1, \quad \sum_{i,j} x_{i,j,8} = 5$$

- **Weekly Study Requirements:**

$$\sum_{i,j} x_{i,j,k} \geq R_k \quad \forall k \in \{1,2,3,4\}$$

$$R_1 = 12,\ R_2 = 12,\ R_3 = 10,\ R_4 = 8$$

- **Lunch and Dinner:**

$$\sum_{j=11}^{13} x_{i,j,5} = 1, \quad \sum_{j=8}^{22} x_{i,j,5} = 1 \quad \forall i$$

$$\sum_{j=17}^{19} x_{i,j,6} = 1, \quad \sum_{j=8}^{22} x_{i,j,6} = 1 \quad \forall i$$

- **Exercise Constraints:**

$$\sum_{i,j} x_{i,j,7} = 3, \quad e_i \geq \sum_{j=8}^{22} x_{i,j,7} \quad \forall i$$

$$e_i + e_{i+1} \leq 1 \quad \forall i \in \{1,\ldots,6\}, \quad e_7 + e_1 \leq 1$$

- **Preallocated Slots:**

$$x_{i,j,k} = 1 \quad \text{for all fixed slots}$$

- **Break Detection:**

$$\sum_k x_{i,j-1,k} - \sum_k x_{i,j,k} \geq 1 - 2(1 - \delta_{i,j}), \quad \sum_k x_{i,j-1,k} - \sum_k x_{i,j,k} \leq \delta_{i,j}$$

Variable Domains:

$$x_{i,j,k},\ s_{i,j,k},\ e_i,\ \delta_{i,j} \in \{0,1\}$$

Solving the ILP in JuMP

The model output the values of all $x_{i,j,k}$. From that, we extracted all the variables which were equal 1, and tabulated the results in the timetable below.

$x_{1,8,7} = 1,\quad x_{1,10,1} = 1,\quad x_{1,11,1} = 1,\quad x_{1,13,5} = 1,\quad x_{1,15,4} = 1,\quad x_{1,16,4} = 1,$
$x_{1,18,6} = 1,\quad x_{1,20,2} = 1,\quad x_{1,21,2} = 1,\quad x_{2,10,2} = 1,\quad x_{2,11,2} = 1,\quad x_{2,12,5} = 1,$
$x_{2,13,1} = 1,\quad x_{2,14,1} = 1,\quad x_{2,16,3} = 1,\quad x_{2,17,3} = 1,\quad x_{2,18,6} = 1,\quad x_{2,20,4} = 1,$
$x_{2,21,4} = 1,\quad x_{3,8,1} = 1,\quad x_{3,9,1} = 1,\quad x_{3,11,5} = 1,\quad x_{3,13,2} = 1,\quad x_{3,14,2} = 1,$
$x_{3,16,3} = 1,\quad x_{3,17,3} = 1,\quad x_{3,18,6} = 1,\quad x_{3,19,9} = 1,\quad x_{3,20,9} = 1,\quad x_{3,21,9} = 1,$
$x_{4,8,4} = 1,\quad x_{4,9,4} = 1,\quad x_{4,11,2} = 1,\quad x_{4,12,2} = 1,\quad x_{4,15,1} = 1,\quad x_{4,16,1} = 1,$
$x_{4,18,6} = 1,\quad x_{4,20,3} = 1,\quad x_{4,21,3} = 1,\quad x_{5,9,3} = 1,\quad x_{5,10,3} = 1,\quad x_{5,12,5} = 1,$
$x_{5,14,1} = 1,\quad x_{5,15,1} = 1,\quad x_{5,17,6} = 1,\quad x_{5,19,4} = 1,\quad x_{5,20,4} = 1,\quad x_{6,8,2} = 1,$
$x_{6,9,2} = 1,\quad x_{6,11,5} = 1,\quad x_{6,13,8} = 1,\quad x_{6,14,8} = 1,\quad x_{6,15,8} = 1,\quad x_{6,16,8} = 1,$
$x_{6,17,8} = 1,\quad x_{6,19,6} = 1,\quad x_{6,21,7} = 1,\quad x_{7,8,3} = 1,\quad x_{7,9,3} = 1,\quad x_{7,11,5} = 1,$
$x_{7,14,1} = 1,\quad x_{7,15,1} = 1,\quad x_{7,17,6} = 1,\quad x_{7,19,2} = 1,\quad x_{7,20,2} = 1$

Time	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00–09:00	EX ($x_{1,8,7}$)		OPT ($x_{3,8,1}$)	HASS ($x_{4,8,4}$)		PNS ($x_{6,8,2}$)	DBA ($x_{7,8,3}$)
09:00–10:00			OPT ($x_{3,9,1}$)	HASS ($x_{4,9,4}$)	DBA ($x_{5,9,3}$)	PNS ($x_{6,9,2}$)	DBA ($x_{7,9,3}$)
10:00–11:00	OPT ($x_{1,10,1}$)	PNS ($x_{2,10,2}$)			DBA ($x_{5,10,3}$)		
11:00–12:00	OPT ($x_{1,11,1}$)	PNS ($x_{2,11,2}$)	Lunch ($x_{3,11,5}$)	PNS ($x_{4,11,2}$)		Lunch ($x_{6,11,5}$)	Lunch ($x_{7,11,5}$)
12:00–13:00		Lunch ($x_{2,12,5}$)		PNS ($x_{4,12,2}$)	Lunch ($x_{5,12,5}$)		
13:00–14:00	Lunch ($x_{1,13,5}$)	OPT ($x_{2,13,1}$)	PNS ($x_{3,13,2}$)			Leisure ($x_{6,13,8}$)	
14:00–15:00		OPT ($x_{2,14,1}$)	PNS ($x_{3,14,2}$)		OPT ($x_{5,14,1}$)	Leisure ($x_{6,14,8}$)	OPT ($x_{7,14,1}$)
15:00–16:00	HASS ($x_{1,15,4}$)			OPT ($x_{4,15,1}$)	OPT ($x_{5,15,1}$)	Leisure ($x_{6,15,8}$)	OPT ($x_{7,15,1}$)
16:00–17:00	HASS ($x_{1,16,4}$)	DBA ($x_{2,16,3}$)	DBA ($x_{3,16,3}$)	OPT ($x_{4,16,1}$)		Leisure ($x_{6,16,8}$)	
17:00–18:00		DBA ($x_{2,17,3}$)	DBA ($x_{3,17,3}$)		Dinner ($x_{5,17,6}$)	Leisure ($x_{6,17,8}$)	Dinner ($x_{7,17,6}$)
18:00–19:00	Dinner ($x_{1,18,6}$)	Dinner ($x_{2,18,6}$)	Dinner ($x_{3,18,6}$)	Dinner ($x_{4,18,6}$)			
19:00–20:00			5th Row ($x_{3,19,9}$)		HASS ($x_{5,19,4}$)	Dinner ($x_{6,19,6}$)	PNS ($x_{7,19,2}$)
20:00–21:00	PNS ($x_{1,20,2}$)	HASS ($x_{2,20,4}$)	5th Row ($x_{3,20,9}$)	DBA ($x_{4,20,3}$)	HASS ($x_{5,20,4}$)		PNS ($x_{7,20,2}$)
21:00–22:00	PNS ($x_{1,21,2}$)	HASS ($x_{2,21,4}$)	5th Row ($x_{3,21,9}$)	DBA ($x_{4,21,3}$)		EX ($x_{6,21,7}$)	
22:00–23:00							

Interpretation of Results

The timetable shows that activities are spaced across the week, and across the day, which aligns with our objective function intention. The constraints set are all adhered to, which allows for a balanced and productive timetable, with time in betweeen activities to work on miscellaneous tasks.

We note that for our problem, there are multiple optimal solutions. For instance, both PNS slots on Sunday 7pm-9pm could be moved to 8pm-10pm, which would incur the same cost.

The model is effective only if constraints are set carefully. For example, if too little activities are scheduled, the activities might become front loaded in the day, because only spacings of 1 hour is incentivised. Thus, this is a limitation of our objective function.

Model Assumptions and Limitations

- **Fixed Time Granularity:** The model discretizes the day into hourly slots (8am to 11pm). This assumes that all activities (meals, study, leisure, etc.) align cleanly with 1-hour blocks, which may not reflect real-world variability (e.g., a lecture that spans 1.5 hours).
- **No Overlapping Activities:** Each time slot allows at most one activity to be scheduled. This assumes that multitasking or passive activities (e.g., listening to a podcast while eating) are not feasible or meaningful in Alex's context.
- **Simplified Meal and Sleep Allocation:** Lunch and dinner are rigidly constrained to fixed windows, and sleep hours are excluded from the model (assumed to occur from 11pm to 8am). This simplifies the schedule but may omit useful flexibility (e.g., power naps or late meals).
- **No Travel or Transition Time:** The model does not account for transition time between activities. In practice, moving between locations (e.g., classroom to gym) requires time, which may compress available scheduling windows.
- **Binary Session Start Logic ($s_{i,j,k}$):** The logic behind session start indicators assumes that all study and leisure blocks must begin at a clean hour and continue for the defined duration. It does not permit staggered starts or early exits.
- **Strict Enforcement of Preferences:** Constraints such as "no exercise on consecutive days" and "exactly one 5-hour leisure session" are enforced strictly, even if slightly more flexible scheduling could result in a better-spaced or more productive week.
- **No Consideration of Activity Interactions:** The model does not account for interactions between activities. For instance, intense exercise may influence the effectiveness of subsequent study slots, or fifth row activities may impose fatigue.

Appendix: Code Snippets

```
using JuMP
using MosekTools

# Initialize model
model = Model(Mosek.Optimizer)

# Index sets
days = 1:7                  # i (days: Mon to Sun)
hours = 8:22                # j (hour slots from 8am to 10pm)
activities = 1:9            # k (activity types)

# Define decision variables
@variable(model, x[days, hours, activities], Bin)   # Activity scheduled
@variable(model, s[days, hours, activities], Bin)   # Activity start
@variable(model, δ[days, hours], Bin)               # Break immediately after activity
@variable(model, e[days], Bin)

# Define Constraints:
# Constraint 1: Each activity (k) can only be scheduled once per day (i) and hour (j)
for i in days, j in hours
    @constraint(model, sum(x[i,j,k] for k in activities) <= 1)
end

# Constraint 2: Each academic subject (k = 1 to 4) ≤ 2 hours per day
for i in days, k in 1:4
    @constraint(model, sum(x[i,j,k] for j in hours) <= 2)
end

# Constraint 3: consecutive hours constraint for study and for leisure

# Start detection constraints for study subjects and leisure (k ∈ {1,2,3,4,8})
for i in days, j in 9:22, k in [1, 2, 3, 4, 8]  # j starts at 9 to use j-1 = 8
    @constraint(model, x[i,j,k] - x[i,j-1,k] >= -1 + 2*s[i,j,k])
    @constraint(model, x[i,j,k] - x[i,j-1,k] <= s[i,j,k])
end
```

```
# Minimum 2-hour study session for each start (k ∈ {1,2,3,4}, j ∈ 8 to 21)
for i in days, j in 8:21, k in 1:4
    @constraint(model, sum(x[i,j+τ,k] for τ in 0:1) >= 2 * s[i,j,k])
end

# Edge case: Prevent starting 2-hour session at hour 22
for i in days, k in 1:4
    @constraint(model, s[i,22,k] == 0)
end

# Edge case: Prevent isolated 1-hour session at hour 8
for i in days, k in 1:4
    @constraint(model, x[i,8,k] <= x[i,9,k])
end

# Minimum 5-hour leisure session for each start (k = 8, i ∈ {6,7}, j ∈ 8 to 18)
for i in 6:7, j in 8:18
    @constraint(model, sum(x[i,j+τ,8] for τ in 0:4) >= 5 * s[i,j,8])
end

# Exactly one leisure session starting on the weekend
@constraint(model, sum(s[i,j,8] for i in 6:7, j in 8:18) == 1)

# Total of exactly 5 leisure hours across the week
@constraint(model, sum(x[i,j,8] for i in days, j in hours) == 5)

# Constraint 4: Required weekly hours per subject
R = Dict(1 => 12, 2 => 12, 3 => 10, 4 => 8)

for k in 1:4
    @constraint(model, sum(x[i,j,k] for i in days, j in hours) >= R[k])
end
```

```
# Constraint 5: Lunch (activity 5) occurs exactly once per day between 11am–2pm (j = 11 to 13)
for i in days
    @constraint(model, sum(x[i,j,5] for j in 11:13) == 1)
end

for i in days
    @constraint(model, sum(x[i,j,5] for j in 8:22) == 1)
end

# Constraint 6: Dinner (activity 6) occurs exactly once per day between 5pm–8pm (j = 17 to 19)
for i in days
    @constraint(model, sum(x[i,j,6] for j in 17:19) == 1)
end

for i in days
    @constraint(model, sum(x[i,j,6] for j in 8:22) == 1)
end

# Constraint 7: Exercise

# Exactly 2 exercise sessions per week (k = 7)
@constraint(model, sum(x[i,j,7] for i in days, j in hours) == 2)

# e[i] must be 1 if any exercise on day i
for i in days
    @constraint(model, e[i] >= sum(x[i,j,7] for j in hours))
end

# No exercise on consecutive days (wrap around from Sunday to Monday)
for i in 1:6
    @constraint(model, e[i] + e[i+1] <= 1)
end
@constraint(model, e[7] + e[1] <= 1)
```

```
# Constraint 8: Preallocated Timetable Slots
@constraint(model, x[1,10,1] == 1)
@constraint(model, x[1,11,1] == 1)
@constraint(model, x[1,15,4] == 1)
@constraint(model, x[1,16,4] == 1)
@constraint(model, x[2,10,2] == 1)
@constraint(model, x[2,11,2] == 1)
@constraint(model, x[2,13,1] == 1)
@constraint(model, x[2,14,1] == 1)
@constraint(model, x[2,16,3] == 1)
@constraint(model, x[2,17,3] == 1)
@constraint(model, x[3,19,9] == 1)
@constraint(model, x[3,20,9] == 1)
@constraint(model, x[3,21,9] == 1)
@constraint(model, x[4,9,4] == 1)
@constraint(model, x[4,11,2] == 1)
@constraint(model, x[4,12,2] == 1)
@constraint(model, x[5,9,3] == 1)
@constraint(model, x[5,10,3] == 1)

# Define Objective Function
# Breaks right after activities (j = 9 to 22 because j-1 must be valid)
for i in days, j in 9:22
    @constraint(model, sum(x[i,j-1,k] for k in 1:9) - sum(x[i,j,k] for k in 1:9) >= 1 - 2*(1 - δ[i,j]))
    @constraint(model, sum(x[i,j-1,k] for k in 1:9) - sum(x[i,j,k] for k in 1:9) <= δ[i,j])
end

# Objective: Maximize total number of breaks
@objective(model, Max, sum(δ[i,j] for i in days, j in 9:22))
```

$\delta_{1,9} + \delta_{1,10} + \delta_{1,11} + \delta_{1,12} + \delta_{1,13} + \delta_{1,14} + \delta_{1,15} + \delta_{1,16} + \delta_{1,17} + \delta_{1,18} + \delta_{1,19} + \delta_{1,20} + \delta_{1,21} + \delta_{1,22} + \delta_{2,9} + \delta_{2,10} + \delta_{2,11} + \delta_{2,12} + \delta_{2,13} + \delta_{2,14} + \delta_{2,15} + \delta_{2,16} + \delta_{2,17} + \delta_{2,18} + \delta_{2,19} + \delta_{2,20} + \delta_{2,21} + \delta_{2,22} + \delta_{3,9} + \delta_{3,10} + [[\ldots \text{38 terms omitted} \ldots]] + \delta_{5,21} + \delta_{5,22} + \delta_{6,9} + \delta_{6,10} + \delta_{6,11} + \delta_{6,12} + \delta_{6,13} + \delta_{6,14} + \delta_{6,15} + \delta_{6,16} + \delta_{6,17} + \delta_{6,18} + \delta_{6,19} + \delta_{6,20} + \delta_{6,21} + \delta_{6,22} + \delta_{7,9} + \delta_{7,10} + \delta_{7,11} + \delta_{7,12} + \delta_{7,13} + \delta_{7,14} + \delta_{7,15} + \delta_{7,16} + \delta_{7,17} + \delta_{7,18} + \delta_{7,19} + \delta_{7,20} + \delta_{7,21} + \delta_{7,22}$

```
optimize!(model)
solution_summary(model)
value.(x) #the decision variable solution
objective_value(model) #the objective variable solution
```

```
Objective of best integer solution : 3.500000000000e+01
Best objective bound               : 3.500000000000e+01
Initial feasible solution objective: Undefined
Construct solution objective       : Not employed
User objective cut value           : Not employed
Number of cuts generated           : 1676
  Number of Gomory cuts            : 272
  Number of CMIR cuts              : 200
  Number of clique cuts            : 248
  Number of implied bound cuts     : 163
  Number of knapsack_cover cuts    : 793
Number of branches                 : 9712
Number of relaxations solved       : 12560
Number of interior point iterations: 5
Number of simplex iterations       : 579469
Time spend presolving the root     : 0.02
Time spend optimizing the root     : 0.04
Mixed integer optimizer terminated. Time: 10.76

Optimizer terminated. Time: 10.77
```

```
for i in days, j in hours, k in activities
    if value(x[i,j,k]) > 0.9
        println("Day $i, Hour $j, Activity $k scheduled")
    end
end
```

```
Day 1, Hour 8, Activity 3 scheduled
Day 1, Hour 9, Activity 3 scheduled
Day 1, Hour 10, Activity 1 scheduled
Day 1, Hour 11, Activity 1 scheduled
Day 1, Hour 13, Activity 5 scheduled
Day 1, Hour 15, Activity 4 scheduled
Day 1, Hour 16, Activity 4 scheduled
Day 1, Hour 18, Activity 6 scheduled
Day 1, Hour 20, Activity 2 scheduled
Day 1, Hour 21, Activity 2 scheduled
Day 2, Hour 8, Activity 9 scheduled
Day 2, Hour 10, Activity 2 scheduled
Day 2, Hour 11, Activity 2 scheduled
Day 2, Hour 12, Activity 5 scheduled
Day 2, Hour 13, Activity 1 scheduled
Day 2, Hour 14, Activity 1 scheduled
Day 2, Hour 16, Activity 3 scheduled
Day 2, Hour 17, Activity 3 scheduled
Day 2, Hour 19, Activity 6 scheduled
Day 2, Hour 21, Activity 4 scheduled
Day 2, Hour 22, Activity 4 scheduled
Day 3, Hour 8, Activity 2 scheduled
Day 3, Hour 9, Activity 2 scheduled
Day 3, Hour 11, Activity 5 scheduled
Day 3, Hour 13, Activity 9 scheduled
Day 3, Hour 15, Activity 7 scheduled
Day 3, Hour 17, Activity 6 scheduled
Day 3, Hour 19, Activity 9 scheduled
Day 3, Hour 20, Activity 9 scheduled
Day 3, Hour 21, Activity 9 scheduled
Day 4, Hour 8, Activity 4 scheduled
Day 4, Hour 9, Activity 4 scheduled
Day 4, Hour 11, Activity 2 scheduled
Day 4, Hour 12, Activity 2 scheduled
```

```
Day 4, Hour 12, Activity 2 scheduled
Day 4, Hour 13, Activity 5 scheduled
Day 4, Hour 15, Activity 3 scheduled
Day 4, Hour 16, Activity 3 scheduled
Day 4, Hour 18, Activity 6 scheduled
Day 4, Hour 20, Activity 1 scheduled
Day 4, Hour 21, Activity 1 scheduled
Day 5, Hour 8, Activity 9 scheduled
Day 5, Hour 9, Activity 3 scheduled
Day 5, Hour 10, Activity 3 scheduled
Day 5, Hour 11, Activity 5 scheduled
Day 5, Hour 13, Activity 1 scheduled
Day 5, Hour 14, Activity 1 scheduled
Day 5, Hour 16, Activity 4 scheduled
Day 5, Hour 17, Activity 4 scheduled
Day 5, Hour 19, Activity 6 scheduled
Day 5, Hour 21, Activity 7 scheduled
Day 6, Hour 8, Activity 3 scheduled
Day 6, Hour 9, Activity 3 scheduled
Day 6, Hour 11, Activity 5 scheduled
Day 6, Hour 13, Activity 1 scheduled
Day 6, Hour 14, Activity 1 scheduled
Day 6, Hour 16, Activity 9 scheduled
Day 6, Hour 18, Activity 6 scheduled
Day 6, Hour 20, Activity 2 scheduled
Day 6, Hour 21, Activity 2 scheduled
Day 7, Hour 8, Activity 1 scheduled
Day 7, Hour 9, Activity 1 scheduled
Day 7, Hour 11, Activity 5 scheduled
Day 7, Hour 13, Activity 8 scheduled
Day 7, Hour 14, Activity 8 scheduled
Day 7, Hour 15, Activity 8 scheduled
Day 7, Hour 16, Activity 8 scheduled
Day 7, Hour 17, Activity 8 scheduled
Day 7, Hour 19, Activity 6 scheduled
Day 7, Hour 20, Activity 2 scheduled
Day 7, Hour 21, Activity 2 scheduled
```