University of Mila
Faculty of Mathematics and Computer Science
1st Year Computer Science,
2025/2026

Practical Work – Introduction to Artificial Intelligence

PW n°3: AI for Time and Task Management

Exercise 1: Smart To-Do List Generator

Given the following Python code, please write it, execute it, and answer the following questions. (Note: **tasks_dataset** is used in all Exercises)

```
# Common Dataset (Use in all exercises)
tasks_dataset = [
    {"task": "Study Math", "category": "Study", "deadline": 2, "importance": 4, "duration": 2},
    {"task": "AI Project", "category": "Project", "deadline": 5, "importance": 5, "duration": 4},
    {"task": "Read Chapter 4", "category": "Study", "deadline": 7, "importance": 3, "duration": 1},
    {"task": "Prepare Presentation", "category": "Project", "deadline": 3, "importance": 4, "duration": 3},
    {"task": "Organize Notes", "category": "Personal", "deadline": 6, "importance": 2, "duration": 1}
]
def assign_priority(task):
    if task["deadline"] <= 2:
        return "High"
    elif task["deadline"] <= 5:
        return "Medium"
    else:
        return "Low"

for task in tasks_dataset:
    print(task["task"], "| Priority:", assign_priority(task))
```

Questions :

1. What does the for loop do in this Explain the condition deadline <= 2.
2. Add a "Very High" priority if deadline = 1.
3. Display the task category as well.
4. Count the number of High priority tasks.
5. Add a new task to the dataset.

Exercise 2: Task Prioritization System

Given the following Python code, please write it, execute it, and answer the following questions.

```
def compute_score(task):
    return task["importance"] - task["deadline"]

tasks_sorted = sorted(tasks_dataset, key=compute_score, reverse=True)

for task in tasks_sorted:
    print(task["task"], "| Score:", compute_score(task))
```

Questions :

1. Why do we use `reverse=True` in the sorting function?
2. Modify the score formula to include the task duration.
3. Display only the task with the maximum score.
4. Sort the tasks from the smallest to the largest score.

Exercise 3: Automatic Study Schedule Generator

Given the following Python code, please write it, execute it, and answer the following questions

```
total_hours = 8

total_duration = sum(task["duration"] for task in tasks_dataset)

for task in tasks_dataset:
    allocated_time = (task["duration"] / total_duration) *
total_hours
    print(task["task"], "->", round(allocated_time, 2), "hours")
```

Questions

1. Calculate the total duration of all tasks in the dataset.
2. Allocate a total of 8 study hours proportionally according to each task's duration.
3. Display the allocated hours for each task.
4. Verify that the sum of allocated hours equals 8 hours.

Exercise 4 (Supplementary): Keyword-Based Task Analyzer

Given the following Python code:

```
keyword = input("Enter a keyword: ")
found = False

for task in tasks_dataset:
    if keyword.lower() in task["task"].lower():
        print("Matched Task:", task["task"])
        found = True

if not found:
    print("No task found.")
```

Questions:

1. Ask the user to input a keyword.
2. Display all tasks that contain the entered keyword.
3. Count and display the number of matching tasks.
4. If the user types "all", display all tasks.
5. Display a message if no task matches the keyword.