ELECTRICAL TEAM TRAINING

TASK 1

TABLE OF CONTENTS

PREFACE

Step into a realm where the vibrant **Minions universe** converges with the captivating world of programming. As the golden rays of **Dr. Nefario's** genius illuminate the horizon, a new era of technological prowess and creative endeavors dawns upon you. Welcome to an apprenticeship like no other, where innovation and imagination intertwine seamlessly.

In this unique voyage, **you are chosen to become Dr. Nefario's apprentice**, entrusted with crafting ingenious solutions for the ambitious dreams of Gru, the mastermind behind countless Minion escapades. Your journey begins with an exciting series of tasks, each intricately woven into the tapestry of Minion adventures.

TASK1.1- Hello Gru

About

Gru showcases a unique and multifaceted ego that evolves throughout time. Initially portrayed as a cunning and ambitious supervillain with a desire to steal the moon, Gru's ego is driven by his need to prove himself as the greatest villain.

Gru has his own dream to steal the moon and set up an enormous digital screen and write his name _'GRU'_ on it, and as Dr. Nefario's apprentice, he asked you to do this job.

Requirement

Just write a C/C++ program that prints the string 'GRU' in the output terminal (console) (Hello World problem).

Output

GRU

Bonus: Draw GRU name with astreics

```
*****    ****     *     *
*        *   *    *     *
*  **    ****     *     *
*   *    *   *    *     *
*****    *    *   ******
```

TASK1.2- Counting Down

About

Following the triumphant mission of thwarting Victor Perkins and restoring the moon to its rightful place, Gru now harbors a fervent dream – to journey to the moon, much like his idol, Neil Armstrong.

Your mission, should you choose to accept it, is to code a program that initiates a countdown for Gru's lunar expedition. The program should efficiently display the countdown as it approaches the launch moment

Requirement

Write a C/C++ program that takes the number to start the countdown from as input and prints the countdown text starting from that number.

Input Example

5

Output

5
4
3
2
1
Blast off to the moon!

Bonus: Add a 1-second delay between each number in the countdown.

TASK1.3- Task Manager

About

While Gru's moon launch preparations are in full swing, it's crucial to stay organized. As a brilliant apprentice, you're tasked with developing a Minions-themed task manager. This program will help keep track of all the essential to-dos, ensuring nothing gets left behind in the lunar excitement.

Requirement

Create a C/C++ program that allows users to add, view, and remove tasks, each task has an id/index and a description. The id is a positive unique integer.

Output Example

This is just an example of what the output could look like. You can change it but make sure that all the features are present and the interface is interactive and user-friendly.

Minions Task Manager
1. Add Task
2. View Tasks
3. Remove Task
4. Exit

Select an option: 1

Enter task description: Prepare moon launch materials
Task added successfully!

Select an option: 1

Enter task description: Check spaceship fuel levels
Task added successfully!

Select an option: 2

Current Tasks:
Task ID: 1
Description: Prepare moon launch materials

Task ID: 2
Description: Check spaceship fuel levels

Select an option: 3

Enter task ID to remove: 2
Task removed successfully!

Select an option: 2

Current Tasks:
Task ID: 1
Description: Prepare moon launch materials

Select an option: 4
Exiting Minions Task Manager. Have a great day!

Bonus: Take it a step further by allowing users to mark tasks as completed, view complete and incomplete tasks, and show the data in a table format.

TASK1.4- Kalman Missile

CHALLENGING

About

Gru's ego takes a unique turn as he finds himself facing a new challenge – a desire to build a fire-and-forget missile _missile that has a targeting system_ to save himself and those he cares about (this is what he claimed but actually he only cares about his dreams), thus, Dr. Nefario orders you to build his dangerous Missile.

To build an **Autonomous** missile of course you need a sensing capability, but you have only two **IMU sensors** one cheap called MPU6050 has an accuracy of 78% the other one called BNo55 has an accuracy of 92%.

When you asked Dr. Nefario what to do, he went on a long monologue about **Sensor Fusion** and how important it is. What you gathered from his long talk is that if you have two sensors that give you the same information with different measurements, each one of them has a specific accuracy, thus, you could fuse these

measurements and generate one measure that has greater accuracy than both of them, by using data fusion algorithm like **Kalman Filter.** (Better sensor measurements, means better feedback to the control system, which means better positioning of the missile).

NOTE: THIS IS AN ACTUAL PROBLEM FACED BY OUR AUTONOMOUS TEAM.

Requirement

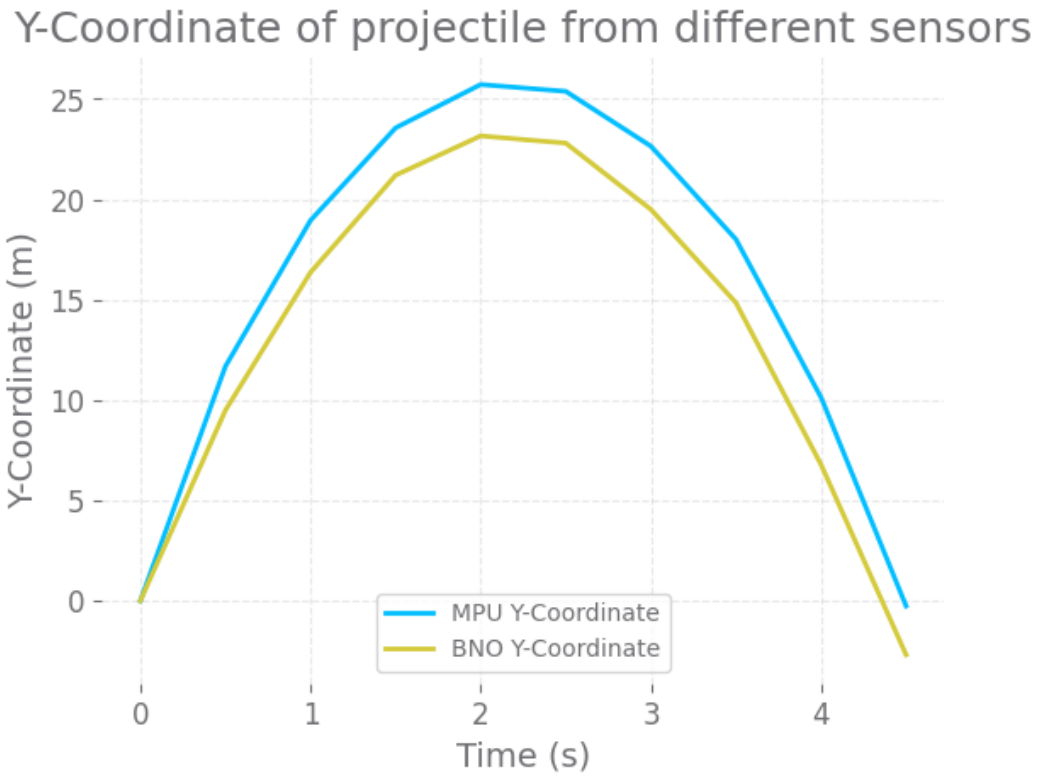
Write a C/C++ program that takes two sensor measurements (just two arrays) and generates only one new array measurement with any averaging method, such as Kalman Filter.

- The two sensors have two different accuracies one has 79% and another one has 92%, (You should consider this information in your output formula).
- IMU sensor outputs the forces in all spatial coordinates x, y, z, but for the sake of simplicity we only work in one dimension which is y.
- Assume two sensors have the same sample frequency, (that means the two arrays have the same number of elements)

Input

There are two static arrays with specific elements that represent the distance of the projectile in the y-axis as extracted from the sensors. You can hardcode them instead of taking them as input from the user.
Here are the arrays data:

```
float mpu6050[10] = {0.0, 11.68, 18.95, 23.56, 25.72, 25.38, 22.65, 18.01, 10.14, -0.26};
float bno55[10]   = {0.0,9.49, 16.36, 21.2, 23.16, 22.8, 19.5, 14.85, 6.79, -2.69};
```



Some data that you could be helpful (but you don't have to use):

- Initial Velocity: 30m/s
- Initial Angle: 46 deg

Output

Generate one single array from the input arrays with the same number of elements (this one array should be a better description of the missile projectile).

- **Output rating:**

1. **Fair:** Used averaging sensor without regard accuracy of each sensor and generated reasonable output
2. **Good:** Used averaging sensor with taking into consideration the accuracy of each sensor and generate reasonable output
3. **Hero:** Used dynamic filter such as Kalman filter and generated reasonable output
4. **Legend:** Used dynamic filter such as Kalman filter and taking into consideration the physics of projectile to infer new information and fuse all this information and generate reasonable output.

Appendix

A IMU

The term IMU stands for “Inertial Measurement Unit,” and we use it to describe a collection of measurement tools. When installed in a device, these tools can capture data about the device’s movement. IMUs contain sensors such as **accelerometers**, **gyroscopes**, and **magnetometers.**

- Article 1: What is IMU? Inertial Measurement Unit Working | Arrow.com | Arrow.com
- Tutorial 1: MPU6050 Module Pinout, Configuration, Features, Arduino Interfacing & Datasheet
- Tutorial 2: Capturing IMU Data with a BNO055 Absolute Orientation Sensor - Projects
- Video 1: (1001) DIY Gimbal | Arduino and MPU6050 Tutorial - YouTube

B Sensor Fusion:

A vehicle could use sensor fusion to fuse information from multiple sensors of the same type as well — for instance, radar. This improves perception by taking advantage of partially overlapping fields of view. As multiple radars observe the environment around a vehicle, more than one sensor will detect objects at the same time. Interpreted through global 360° perception software, detections from those multiple sensors can be overlapped or fused, increasing the detection probability and reliability of objects around the vehicle and yielding a more accurate and reliable representation of the environment.

- Video1: https://www.youtube.com/watch?v=HBXTYNJxMho&pp=ygUNc2Vuc29yIGZ1c2lvbg%3D%3D
- Article 1: What Is Sensor Fusion?
- Video2 (recommended): https://www.youtube.com/watch?v=0rlvvYgmTvI&pp=ygUNc2Vuc29yIGZ1c2lvbg%3D%3D

C Kalman Filter:

The Kalman filter was invented by **Rudolf Emil Kálmán** to solve this sort of problem in a mathematically optimal way. Its first use was on the **Apollo missions to the moon**, and since then it has been used in an enormous variety of domains. Kalman filters are used in aircraft, submarines, and cruise missiles. Wall **Street uses them to track the market**. They are used in **robots**, **IoT (Internet of Things) sensors**, and laboratory instruments. **Chemical plants** use them to control and monitor reactions. They are used to perform **medical imaging** and to **remove noise from cardiac signals**. If it involves a sensor and/or time-series data, a Kalman filter or a close relative to the Kalman filter is usually involved.

- Video1: (1001) Kalman Filter for Beginners - YouTube
- Article1: Kalman Filter Definition | DeepAI

NOTES

- Your code should be clean, well-commented, and easy to follow up.
- Make sure your code compiles successfully without any errors.
- Use suitable names for variables and functions. Don't use vague names like (x, y, etc.)
- Don't to write redundant code. If you find that your code is repetitive then you should probably refactor your code. Maybe the repetitive code should be in a function or needs to be inside a loop.
- Implementing any of the bonuses isn't required but is welcomed and is taken into consideration.
- Implementing extra features that aren't mentioned is welcomed but should be accompanied by thorough and well-written documentation for them.

SUBMISSION

- Submit a separate .c, .cpp or .zip file for each task solution
- Each task solution should follow the following convention:
  If your problem solution consists of a single file then it should be named yourPhoneNumber_Task1_1.c or yourPhoneNumber_Task1_1.cpp
  If your problem solution consists of multiple files then zip them up into 1 file named yourPhoneNumber_Task1_1.zip
- The phone number should be the one you submitted in the form
- Files format example:
  01xxxxxxxx_Task1_1.c
  01xxxxxxxx_Task1_2.c
  01xxxxxxxx_Task1_3.cpp
  01xxxxxxxx_Task1_4.zip
- Submit your solutions to this form: [FORM LINK]
- DEADLINE **Monday 14/8 11:59PM**