ACM - S1 RECRUITMENT TASKS

Every recruitment registrant must complete these standard tasks. These tasks are crucial for club membership. The initial task involves setting up a GitHub account and repository, with each following task requiring separate file uploads to this repository.

Prior Instructions :

- **All submitted content must be original and not copied from existing works.**
- **You are free to use any programming language of your choice.**
- **Only submissions made before the deadline will be considered.**
- **Changes to the GitHub repository after the deadline are discouraged and will not be accepted.**
- **Prioritize using GitBash for Git operations instead of direct uploading and through VS-Code.**
- **Ensure your code is well-documented and easy to understand.**
- **Include a README file with instructions on how to run your code.**
- **Test your code thoroughly before submission.**
- **Adhere to any specific formatting or organizational guidelines provided.**

- **If using external libraries, provide a list of dependencies and instructions for installation.**
- **Submissions should include any necessary files to run the code without additional setup.**

COMMON TASKS

1.	**Kickstart Your ACM Recruitment Journey** Step 1: Create a GitHub Account • If you don't already have a GitHub account, please create one. • Create a new, empty repository named ACMRecruitment-<yourname>. • Upload all your completed tasks to this repository. • You will need to submit the link to this repository when you finish all the tasks. Please create accounts on the following platforms if you don't already have them. Once created, add your UserIDs to your GitHub repository: 1. LinkedIn 2. DevPost 3. LeetCode 4. HackerRank
2.	**Hands-On Programming Task** Complete the CS-50x 2024 up to lecture 6, playlist - [YouTube]
3.	**Project Euler** Project Euler challenges participants with mathematical problems that require programming to solve. Participants must solve any 5 of the first 10 questions from the problems archive using any programming language and upload the codes along with screenshots of outputs into the GitRepo - [Project Euler]

4.	**Why Did the Calculator Apply for a Job?** **Because It Wanted to Crunch More Numbers and Make Fewer Mistakes!** **Calculator Tool (Object Oriented)** Develop an object-oriented command-line calculator that handles a range of mathematical operations and respects the order of operations. Requirements: 1. Basic Operations: Implement addition, subtraction, multiplication, and division. 2. Advanced Operations: Include square roots and exponents. 3. Order of Operations: Ensure the calculator respects the order of operations (Parentheses, Exponents, Multiplication, Division (left to right), Addition, and Subtraction (left to right)). 4. Parentheses Handling: Support for parentheses to dictate the order of operations. 5. Error Handling: Provide meaningful error messages for invalid inputs, such as division by zero or malformed expressions. 6. Object-Oriented Design: Structure your code using classes and objects. Implement separate classes or methods for different functionalities (e.g., operations, expression parsing). 7. User Input: Allow the user to input mathematical expressions in a single line, e.g., "(2+3)*4^2". Upload the code along with the screenshots of output into the GitHub Repository
5.	**Why do passwords never get lost? Because they always hash out their issues!** Password validation ensures that only secure passwords are accepted, by checking for specific criteria that enhance security Define a function that takes a password as a parameter and validates it; Validation conditions : • Length 8 • Contains at least one capital letter and one small letter • Contains at least one special character • Should not start with a number or special character • passwords = ["A1b#cD3e", "Xy4$Zz7!", "P@ssw0rd", "M!n3r4L^", "T7r$eN8f"]; I don't want these passwords to be accepted

	• The function should return an error message if the password is not valid
6.	**Why did the computer get kicked out of the Tic-Tac-Toe game?** **It kept trying to play in binary!** **Build a Tic-Tac-Toe Game** Develop a command-line Tic-Tac-Toe game where a player competes against a computer that makes random moves. The game should handle invalid moves and ensure smooth gameplay. Requirements: • Implement a single-player Tic-Tac-Toe game where the player plays against the computer. • The computer should make random moves. • The player should input their moves via the command line. • If the player attempts to make an invalid move (e.g., selecting an already occupied position), the game should display an "Invalid Move" message. • The game should correctly manage win conditions and draw scenarios. Upload the code along with the screenshots of output into the GitHub Repository
7.	**Why don't programmers need glasses? Because they can C in binary, and when it comes to bugs, they just shift them out of sight!** **Solve the following Binary Maze Challenge to test your understanding of Binary.[No need of a program]** – **Problem**
8.	**OverTheWire Bandit** It is a game that is designed to teach players the basics of Linux command line and security concepts. It provides practical experience in navigating the file system, manipulating files, and understanding user permissions. Complete up to level 20 and maintain documentation. [overthewire_bandit]

Contact us for Queries regarding common tasks

Akhil (9400646772), Lokesh (7095419591), Gowri (8547344344), Madan (9392532515)

Join our Discord channel for quick assistance - DISCORD

SIG SPECIFIC TASKS

These tasks are SIG-specific and voluntary, customized to match your chosen Special Interest Group within the club. By finishing these particular tasks, you can exhibit your passion and proficiency, greatly improving your chances of joining your preferred SIG. Embrace the opportunity and display your talents!

SIG - AI

Create a Presentation on AI Concepts

Step 0:
Complete the CS50x 2024 Artificial Intelligence Lecture and save your work- [YouTube]

Step 1:
Explore and present fundamental AI concepts through a video presentation. Your task is to clearly explain key topics in AI and machine learning, showcasing your understanding with detailed explanations and insights.

- Slide 1: What is the working Principle of AI and What is a Model?
- Slide 2: What is a Classifier? Define and explain a few classifiers you know.
- Slide 3: Clearly explain Supervised vs Unsupervised Learning
- Slide 4: How ChatGPT and Gemini works

The content should not be AI-generated, a video should be there explaining the concepts with your voiceover, upload it on YouTube and put the link in the Github Tasks Repo

Power of Preprocessing

Why did the data analyst break up with the loan dataset? Because it was full of missing values and bad dependencies! Get ready to clean up and prep your Loan Prediction Dataset with a series of fun and essential preprocessing steps.

Loan Prediction Dataset Preprocessing Tasks

Step 0: Download the Dataset
- Download the dataset from the following source: Loan Prediction Problem

Step 1: Loading the Data
- Load the Loan Prediction Dataset into a pandas DataFrame from the CSV file. Save the DataFrame as df.

Step 2: Inspecting the Data
- Display the first 5 rows of the DataFrame.
- Display the last 5 rows of the DataFrame.

Step 3: Handling Missing Values
- Find all the names of columns having missing values.
- Find the number of missing values in each of the columns identified in the above step.

Step 4: Finding Non-Numerical Values
- List all the columns which contain non-numerical entries.

Step 5: Filtering
- Find all the Loan IDs of people who are not graduates.
- Find all the Loan IDs of people who have an Applicant Income of less than 5000.
- Find all the Loan IDs where the person is unmarried and has dependents '3+'.
- Print all the rows which have Dependents '3+'.
- Drop the rows having null/missing values.

Step 6: Drop Unwanted Columns
- Drop the column Loan_ID from the DataFrame.

Note: use Jupyter Notebooks to perform all the above steps

Predicting the Output

You know what, AI is nothing but a Line in your hands try experiencing it.
You are given a dataset in a list where each entry consists of three numbers: a, b, and value. The first two numbers (a and b) are the inputs, and the third number (value) is the output associated with these inputs.

Your task is to:

1. Store the Dataset: Store the data in a suitable data structure (e.g., a list or a NumPy array).
2. Develop the Logic: Determine the relationship between a, b, and value using a method of your choice.

3. Create the Function: Write a function that takes two input values, a and b, and returns the corresponding output value based on the relationship you determined.

data = [[0.0, 0.0, 0.0], [0.5, 1.5, 23.4], [1.2, 2.3, 45.6], [1.8, 3.7, 12.1], [2.4, 4.2, 78.9], [2.9, 5.1, 34.5], [3.5, 6.4, 56.7], [4.1, 7.8, 67.8], [4.7, 8.5, 89.0], [5.2, 9.1, 12.3], [5.8, 1.0, 45.6], [6.3, 2.4, 78.9], [6.9, 3.1, 34.5], [7.4, 4.6, 56.7], [8.0, 5.2, 67.8], [8.6, 6.8, 89.0], [9.1, 7.3, 12.3], [9.7, 8.9, 45.6], [10.0, 9.0, 78.9], [10.5, 0.5, 34.5]]

Upload the code along with the screenshots of output into the GitHub Repository

API Integration:

Why did the API cross the road? To fetch data from the other side!

Dive into the world of APIs by integrating and extracting valuable information from a selected service. Your challenge is to harness the power of an API to fetch and display crucial data, showcasing your ability to interact with external data sources.

Task: Use any one of the following two APIs from RapidAPI to make requests and print specific responses:

1. Open Weather API:
   - Fetch the current weather for a city.
   - Print: City Name, Temperature, Weather Description, Wind Speed, Humidity, and place coordinates.

1. WhatsApp Number Validator API:
   - Validate a WhatsApp number.
   - Print: Number, Validity (True/False)

Upload the code along with the screenshots of output into the GitHub Repository

Note: The work should reflect your skills and originality. It should not be a direct copy of any existing resource or publicly available projects. And can contact us anytime regarding solving the mentioned tasks.

Queries: Lokesh (7095419591), Madan (9392532515)

SIG - WEB

Build a Personal Portfolio Website:

- Create a responsive personal portfolio website using HTML, CSS, and JavaScript.
- The website should include sections such as About Me, Projects, Skills, and Contact.
- Implement smooth scrolling and a dynamic project gallery.

Develop a To-Do List Application:

- Create a to-do list application using React.js.
- The app should allow users to add, edit, delete, and mark tasks as completed.
- Implement local storage to save tasks between sessions.

Build a Simple Smart Contract:

- Write and deploy a simple smart contract on the Ethereum blockchain using Solidity.
- The smart contract should manage a basic token with functionalities like minting, transferring, and checking balances.
- Test the smart contract using Remix IDE and deploy it to the Ropsten test network.

Deploy a Web Application on AWS:

- Deploy a simple web application (e.g., a portfolio site or a blog) on AWS.
- Use services like EC2 for the server, S3 for storage, and Route 53 for DNS management.
- Ensure the application is secure with proper IAM roles and policies.

Feel free to ping the mentor for pointers and resources ;)

Queries: Akhil (9400646772)

SIG - CYBER

Basic Web Security

Finish the following labs and upload the Screenshots

Set-1

Set-2

Set-3

Use this knowledge to perform a Secure Code Review:

- Perform a secure code review on a given piece of software (e.g., a small web application).
- Identify security issues in the code, such as improper input validation, insecure data storage, and insufficient error handling.
- Provide a detailed report with recommendations for fixing the identified issues.

Cryptography

Broadcasts are bad □ :

Plain RSA was too easy apparently, try this variation of RSA. (A guy just told me that the Chinese are too smart XD) – chall

Crack the Code

Bob is facing a new challenge that involves deciphering a message to obtain the cipher text. The cipher text he needs to find has exactly 28 letters. Here are some crucial clues to get started:

- **The first two letters of the cipher text are 'HL'.**
- **The last two letters of the cipher text are 'BW'.**
- **A mysterious grid is provided as part of the challenge.**

link by1 link2

Forensics

What is there inside the file? (Do not use tools to crack, use a script to crack the given zip) – link

Serene Escape □:

One of our members has found a place so serene that they

practically live there. Your mission is to uncover the name of the dam associated with this tranquil reservoir. - link

Queries: Niranjan(7736242443), Akhil (9400646772)

SIG - CP

Write two programs in any language of your choice: one for reversing a string and another for reversing an integer without using built-in reverse functions.

Solve the Following LeetCode questions and submit the acceptance screenshots in GitHub Repo along with userID - (LEETCODE)

Implement a program for constructing a LinkedList with operations to Insert, Delete, and Search elements.

NOTE: ALL THE CODES MUST BE UPLOADED TO THE TASKS REPOSITORY IN GITHUB ALONG WITH THE COMMON TASKS

Queries: Gowri (8547344344), Nanditha (8714480575)

SIG - GLITCH

2D Platformer

Objective: Create a self-contained 2D platformer where the player navigates a character through various platforms, avoids obstacles, collects items, and reaches the end goal. The project should be unique and not a direct replication of any YouTube tutorial.

Requirements:

1. Design a simple platformer-level layout, including various platforms, obstacles, and a goal
2. The main character must be able to move left, right, and jump.

3. Include a scoring system to track collected items.
4. Implement basic physics for character movement and interactions.
5. Create a simple start screen with a play button.
6. Create an end screen that displays the player's score and a replay button.
7. Provide a brief description of the game and its mechanics, including instructions on how to play the game.

Optional Objectives (at least one must be fulfilled):
- Create a main character sprite by hand.
- Design a collectible item (e.g., coins, stars) by hand.
- Create a simple background or tileset by hand.
- Create a walk-and-jump animation for the character by hand.
- Create an idle animation by hand.
- Design one type of obstacle (e.g., spikes, enemies) by hand.

3D FPP Game

Objective: Create a self-contained 3D third-person shooter where the player controls a character to navigate a level, shoot at targets, avoid obstacles, and reach a checkpoint. The project should be unique and not a direct replication of any YouTube tutorial.

Requirements:

1. Design a simple level layout, including various obstacles, targets, and a checkpoint (can use Unity Asset Store assets).
2. The character must be able to move, aim, and shoot.
3. Implement basic aiming and shooting mechanics.
4. Include a scoring system to track targets hit.
5. Implement basic physics for character movement and interactions.
6. Create a simple start screen with a play button.
7. Create an end screen that displays the player's score and a replay button.
8. Provide a brief description of the game and its mechanics, including instructions on how to play the game.

Optional Objectives (at least one must be fulfilled):
- Create a main character model by hand.
- Design a weapon model by hand.
- Create a simple environmental asset (e.g., a tree, crate) by hand.
- Create a shooting and reloading animation for the character by hand.
- Create a running or walking animation for the character by hand.
- Create an idle animation by hand.
- Design one type of enemy or target by hand.

Tools:
- Use Unity for development.
- Utilize Unity Asset Store for additional assets if needed.

Submission:

- Submit the Unity project files.
- Provide a short video (1-2 minutes) showcasing the gameplay and features.

Resources for Your Task

To help you with your project, here are some valuable resources:

Unity Learning Platform: Access a wide range of tutorials and learning materials to get you started with Unity development.

https://learn.unity.com/

Unity Asset Store: Find a variety of assets including models, animations, and scripts that you can use in your project.

https://assetstore.unity.com/

Note: The project should reflect your creativity and originality. It should not be a direct copy of any existing YouTube tutorial or publicly available project.

Queries: Hari(+91 91882 84248), Surya(+91 89216 88445)

GUIDELINES :

Submission Requirements:

1. **SignUp To Github if not yet, create a repository named ACM Recruitment-<your_name>**
2. **Create a folder for each task and upload the code and content related to that task in that folder**
3. **Create a separate folder for SIG Specific Tasks and all the respected tasks should be pushed into that folder**
4. **You need to submit the Github Repository Link during the time of submission**

- **Code Files**: Ensure that your code is clean, well-organized, and commented where necessary. Follow best practices for coding and documentation.
- **Screenshots and Documentation**: For tasks requiring screenshots (e.g., HackerRank problems), make sure to include them in your repository along with any necessary documentation.

Quality and Clarity:

- **Task Accuracy**: Verify that your solutions and implementations meet all the specified requirements and constraints.
- **Error Handling**: Implement appropriate error handling and validation in your code to manage edge cases and invalid inputs.
- **Object-Oriented Design**: For tasks requiring object-oriented programming, use classes and methods effectively to structure your code.

Deadlines and Communication:

- **Timeliness**: Complete and upload your tasks before the deadline and the **Submission Link will be mailed later**
- **Queries**: If you have any questions or face issues, reach out to the designated contacts for each SIG for assistance.

Submission Deadline: 15/09/2024

The submission link will be sent through the mail

– ACM Student CHAPTER, Amritapuri