TechnoJam Tasks

This document consists of various tasks which will help you to begin your journey in the programming world and move one step ahead to be part of Team TechnoJam.

Submission Guidelines

1. Create a github repository with the name ‘technojam-task’.
2. Make folder for respective task with name task_#, where # is a task number.
3. Each task must consist of a README.md file where you have to describe how you approached the task and the difficulty you have faced while solving the task.
4. Only submissions made through the github will be considered.
5. After completing all the above steps fill out **this form**.

Example Submission Repository: https://github.com/ayush0x00/technojam-task

Refer the below table to know the minimum points/criteria required to get qualified for next round based on the year you are currently in. The further selection will be made on a **percentile basis** (the more number of tasks you’ll solve, the better will be the chances of you getting selected). Selected students will be notified via email.

Make sure you don’t get sidetracked into learning a whole language/framework/tech-stack for completing a task. You only have to learn the necessary things which are required to complete the task. Some of the resources are listed below to make your life easier.

- StackOverflow
- Getting started with android
- Geeks For Geeks
- W3School

Also directly copying and pasting code will make that submission invalid, and no points will be counted. The last date for task submission is **22nd March 2021**, which can be extended in case of clashes with ETE. For any further query/doubt feel free to mail us at **technojam@galgotiasuniversity.edu.in.**

Mandatory tasks are compulsory for all years having 0 pt weightage. For rest we have 1 pt, 2 pt and 3 pt for easy, intermediate and hard level questions respectively.

Year	**Min. points required**	**Necessary conditions**
1st	**3 pt**	**-**
2nd	**5 pt**	**At least 1 intermediate tasks required**
3rd	**7 pt**	**At least 1 hard and 1 intermediate required**

Tasks marked with * are mandatory.

1. ***** All these questions on leetcode
    - https://leetcode.com/problems/xor-operation-in-an-array/
    - https://leetcode.com/problems/lucky-numbers-in-a-matrix/
    - https://leetcode.com/problems/minimum-time-visiting-all-points/
    - https://leetcode.com/problems/merge-strings-alternately/
    - https://leetcode.com/problems/unique-morse-code-words/

2. Send at least one meaningful PR on any of your favourite repo in TJ Github Organisation **(Intermediate)**

3. Competitive Programming **(easy)**
    - Complete all the 5 questions from this repository. https://github.com/shreyanshdeep/TechnoJam-CP-Task/tree/main/Easy

4. Competitive Programming **(Intermediate)**
    - Complete all the 3 questions from this repository. https://github.com/shreyanshdeep/TechnoJam-CP-Task/tree/main/Medium

5. Competitive Programming **(hard)**
    - Complete the given question here: https://github.com/shreyanshdeep/TechnoJam-CP-Task/tree/main/Hard

6. Design a desktop application using python to validate an email. **(Intermediate)**
    - Use tkinter for designing UI.
    - Use regex for validating user input as a valid email address or not.

7. Implementation and theory of the LOOK algorithm in your desired language. **(hard)**

8. Build an android app named languageInfo using Kotlin. **(Intermediate)**
    - On main screen display the scrollable list of programming languages(You can use listView/RecyclerView to achieve this)
    - When user clicks on any list item a new screen should appear displaying the basic info about that language.(You can use intent to navigate between screens and textView to display info)

9. Build a todo list for routine tasks using JavaScript, html and css. Supporting database CRUD operations. **(hard)**
    - It must be a single page webApp, displaying a list of todos with following capabilities.
        i. add new items
        ii. delete single/multiple items
        iii. update items
    - Everything must synchronize with the database and the page need not to be reloaded each time to see newly added items.
    - Above points are mandatory for completion of this task, rest you are free to use your imagination.

10. Build a web application that counts the number of words in a paragraph. **(easy)**
    - Allows the user to pass a set of lines as input.
    - Displays the count of the words available in input as output.

11. Design a birthday card in Android Studio using XML. **(easy)**
    - It should contain a single screen displaying the birthday card.
    - Use different layouts and components like(textView, imageView, RelativeLayout etc.) to make it attractive and stand apart.

12. Cyber Security **(easy)**
    - Create a phishing web page of https://www.facebook.com/ . Credentials entered by the target should be available on the host machine.

- Make an android payload (.apk file) to dump target’s contact info remotely, using metasploit.

13. Twitter bot **(hard)**
    - Create a twitter bot, using selenium and without using twitter API, which will
        i. Read tweets, separated by a new line, from an input file(choose input file as per your convenience).
        ii. Add a random image from the desktop, and then tweet it.
    - All the tweets should be made in one session only ie. the user should log in only once in his/her twitter account, make all the tweets as specified above and then log out when all the tweets from the input file are consumed. All the tweets should be made in a single tab of the web browser.

14. There are two games including dice that you can play. In the main game, you move two dice together and get the dollar sum comparable to the result of the rolls. In the subsequent game, you move one dice and acquire the dollar sum identical to the square of that value. Which game has the higher anticipated worth and why?**(Intermediate)**