**NUI Galway**

**Programming and Tools for AI (CT5132/CT5148)**

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Preeti Jagdish Sajjan | 19232625 |
| Sampritha Hassan Manjunath | 19232922 |
| Swati. | 19233301 |

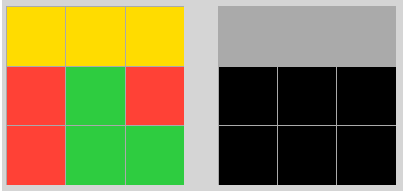
**List of tasks:**

1. 25d8a9c8.json
2. 2dee498d.json
3. d631b094.json
4. c59eb873.json
5. 3ac3eb23.json
6. d10ecb37.json

**GitHub Link:** [**https://github.com/samprithahm/ARC.git**](https://github.com/samprithahm/ARC.git)

|  |
| --- |
| **Task Description:** |

**Task 1:** **25d8a9c8.json**



The task takes the input of 3\*3 grid with random colors filling in. The output of this task is a 3\*3 grid with gray and black colors where a row is colored gray only if all the three consecutive boxes in a row of input grid are of same color else will be colored black.

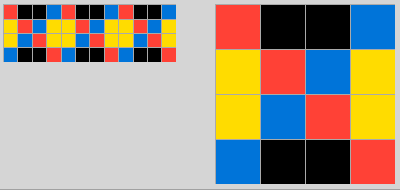
The solve method carries out the bellow steps:

* Check if all the boxes in a row have same color.
* If true then fill the row with gray color in output grid else color it black.
* Iterate the above logic over the rows

A statement:

* From Training set, tested with 2dee498d.json, 1e0a9b12.json, f2829549.json, e50d258f.json and 2bcee788.json and verified successfully
* From Evaluation set, tested with 1c56ad9f.json, 4aab4007.json, 4e469f39.json, 5b6cbef5.json and 992798f6.json and verified successfully.

**Task 2:** **2dee498d.json**



The task takes the input of m\*n grid. The output of this task is a grid of (m\*(n/3)) which is (1/3)rd of the input.

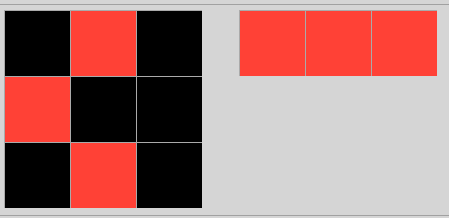
The solve method carries out the bellow steps:

* Divide the input grip by 3
* Assign the first (1/3)rd. division to output grid

A statement:

* From Training set, tested with 25d8a9c8.json, 1e32b0e9.json, 57aa92db.json, 7837ac64.json and 73182012.json and verified successfully
* From Evaluation set, tested with 13713586.json, ac605cbb.json, bb52a14b.json, d492a647.json and fafd9572.json and verified successfully.

**Task 3:** **d631b094.json**



The task takes the input of m\*n grid. The output of this task is a grid of m

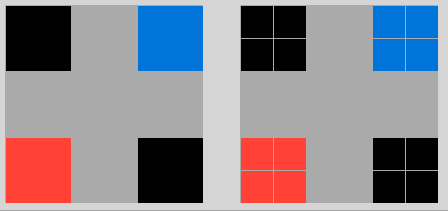
The solve method carries out the bellow steps:

* The input grid is parsed through for non-zero numbers.
* Each non-zero element in the list is added to the output
* The output contains all the non-zero elements in the input

A statement:

* From Training set, tested with 00d62c1b.json, 25d8a9c8.json, 0d3d703e.json, f2829549.json and 1e0a9b12.json and verified successfully
* From Evaluation set, tested with 0a2355a6.json, ac605cbb.json, 4aab4007.json, 4e469f39.json and 0bb8deee.json and verified successfully.

**Task 4:** **c59eb873.json**



The task takes the input of m\*n grid. The output of this task is a grid of (2m \*2n)

The solve method carries out the bellow steps:

* Each element in index “I,j” the input grid is copied and duplicated to “i+1,j+1” position and stored in a list
* Each list is the again duplicated and stored in the output list
* Output is a grid which contains input grid duplicated both row ways and column ways

A statement:

* From Training set, tested with 1b2d62fb.json, 3aa6fb7a.json, 1e0a9b12.json, f2829549.json and 3bd67248.json and verified successfully
* From Evaluation set, tested with 0f63c0b9.json, 1c0d0a4b.json, bb52a14b.json, 0b17323b.json and 2a5f8217.json and verified successfully.

**Task 5:** **3ac3eb23.json**

<img> Input output

(Verbal Description)

(Method Description)

(A Statement)

**Task 6:** **d10ecb37.json**

<img> Input output

(Verbal Description)

(Method Description)

(A Statement)

**Summary:**

Python comes with around 100 modules available to import allowing user to make use of data and functionality from a module which isn’t the current one.

Libraries/Modules used:

* **sys**: The **sys** module provides a set of functions that provide crucial information about how your Python script is interacting with the host system. One of its features which we are using is a command-line option which provides the command line to execute our script.
* **os:** The **OS** module is a part of the standard library or stdlib, within Python 3. It provides functionality allowing the user to interface with the underlying operating system that Python is running on. We are using it in our script to navigate across directory fetching the intended JSON file from the training folder.
* **json:** JSON (**J**ava**S**cript **O**bject **N**otation) is a popular data format used for representing structured data. Python has a built-in package called JSON, which will be useful when working with JSON data. We are using the load functionality of this package to load the intended JSON to our script.

Commonalities: All the above provided solutions use common\_utilities script under Utils folder for reading of file and loading the JSON data to the scripts.

Differences: The underlying logic within each solve method is different.

**Contributions:**

|  |  |  |
| --- | --- | --- |
| Sl. No | Name | Contribution |
| 1 | **Preeti Jagadish Sajjan** | 1. Tasks:   25d8a9c8.json  2dee498d.json   1. Report Template creation 2. Writing summary of each of above-mentioned task |
| 2 | **Sampritha Hassan Manjunath** | 1. Tasks:   d631b094.json  c59eb873.json   1. GitHub and common utilities creation 2. Writing summary of each of above-mentioned task |
| 3 | **Swati** | 1. Tasks:   d631b094.json  8be77c9e.json   1. Readme.md 2. Writing summary of each of above-mentioned task |