# Presto Backend Engineering - Test - Part 1

## Instructions :

1. Write fully functional programs in a programming language of your choice.
2. Write each program in its own file, so that each solution can be separately compiled and / or executed.
    a. Name the file based on the question number, like (in case of Ruby) **`program_1.rb, program_2.rb`**
3. Tests are **absolutely essential**. For every program, write unit tests using a tool / framework of your choice.
    a. Write the tests in a separate file. Name this file also based on the question number, like (in case of Ruby) : **`program_1_test.rb, program_2_test.rb`**
4. If you are not familiar with any testing tool, then write the test cases in plain English. The intent is to see what all test cases you can come up with.
5. Feel free to use the internet for references.

## Part I : General Programming

1. Write a function named **`squarer`** which accepts an array of integers and returns an array with the squares of those integers. Do think of all the possible inputs, both good and bad, that might be passed to this function and make sure the function gracefully handles all those inputs.

2. Write a function **`printer`** which accepts an integer number and prints all the numbers from 1 to that given number but for every number that is a multiple of 3 it prints “Presto” and for every number that is a multiple of 5 it prints “Apps” and for every number that is a multiple of both 3 and 5 it prints “Presto Apps”.

3. Write a class named `DataPacket` which has two attributes :

    `array_data` - of type Array
    `hash_data` - of type Hash (HashMap / Associative Array / Dictionary)

    Write custom setters for both of these attributes so that :

    `array_data` can be set by providing a CSV (Comma Separated Value) string
    `hash_data` can be set by providing a JSON string

    Example (In Ruby Syntax) :

```
# Create an object
d  = DataPacket.new

# Set the value of the array_data attribute by assigning a CSV String
d.array_data = '1,2,3,4'

#After this when we read the value of array_data it should be an array of Integers
d.array_data should be an Array like [1, 2, 3, 4]


# Set the value of the hash_data by assigning a JSON String
```

```
d.hash_data = '{"a": 1, "b": 2, "c": {"d": 3, "e": 4}}'

# After this when we read the value of hash_data it should be a Hash
d.hash_data should be a Hash like

      {
        'a' => 1,
        'b' => 2,
        'c' => {
            'd' => 3,
            'e' => 4
        }
      }
```

4. Write a method named **`interval_iterator`** which will take the following parameters :

   `start_date - A Date string in YYYY-MM-DD format`
   `end_date - A Date string in YYYY-MM-DD format`
   `interval - A string - could be “monthly” or “weekly”`

   And it should return an array of start dates of every interval between the given start date and end date. Start date of an interval is :

   a. Beginning of the month when `interval` is `“monthly”`
   b. Beginning of the week when `interval` is `“weekly”` (Can be Sunday / Monday. Either is fine)

   Examples :

| **Method Invocation** | **Output** |
|---|---|
| `interval_iterator(“01-01-2016”, “31-05-2016”, “monthly”)` | `[“01-01-2016”, “01-02-2016”, “01-03-2016”, “01-04-2016”, “01-05-2016”]` |
| `interval_iterator(“15-01-2016”, “20-05-2016”, “monthly”)` | `[“01-01-2016”, “01-02-2016”, “01-03-2016”, “01-04-2016”, “01-05-2016”]` |
| `interval_iterator(“15-01-2016”, “20-02-2016”, “weekly”)` | `[“18-01-2016”, “25-01-2016”, “01-02-2016”, “08-02-2016”, “15-02-2016”]` |

5. Write a function **`pattern_printer`** which will accept an integer value and print the diamond pattern using the *** (star / asterisk)** as per the examples shown below :

   Ex : For input 3, it should print this pattern :

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

   Ex : For input 6, it should print this pattern :

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