**1. Solving problems using array functions on rest countries data**

**a) Get all the countries from Asia continent /region using Filter function**

To get all the countries from the Asia continent/region using the filter function on a dataset of countries, you can follow these steps:

First, you need to have access to the rest countries data, either by fetching it from an API or by having it available as an array of objects in your code.

Use the filter function to filter out the countries that belong to the Asia continent/region based on the appropriate property in the dataset. In most datasets, you will find a property called "region" or "continent" that contains this information.

Here's an examplecode snippet assuming you have an array of country objects with a "region" property:

// Sample array of country objects

const countries = [

{ name: "India", region: "Asia" },

{ name: "China", region: "Asia" },

{ name: "Japan", region: "Asia" },

{ name: "France", region: "Europe" },

// Add more country objects here...

];

// Use the filter function to get countries from Asia

const asianCountries = countries.filter(country => country.region === "Asia");

console.log(asianCountries);

In this example, asianCountries will contain an array of objects representing countries from the Asia continent/region.

Make sure to replace the sample data with the actual data from the rest countries API or any other data source you are using.

**b) Get all the countries with a population of less than 2 lakhs using Filter function**

To get all the countries with a population of less than 2 lakhs (200,000) using the filter function on a dataset of countries, you can follow these steps:

Assuming you have access to the rest countries data as an array of country objects, you can filter the countries based on their population property.

Use the filter function to filter out the countries that have a population less than 2 lakhs.

Here's an example code snippet assuming you have an array of country objects with a "population" property:

// Sample array of country objects with population data

const countries = [

{ name: "India", population: 1380000000 }, // 1.38 billion

{ name: "China", population: 1444216107 }, // 1.44 billion

{ name: "Japan", population: 126010000 }, // 126 million

{ name: "France", population: 67293000 }, // 67.29 million

// Add more country objects here...

];

// Use the filter function to get countries with a population less than 2 lakhs

const countriesWithLowPopulation = countries.filter(country => country.population < 200000);

console.log(countriesWithLowPopulation);

In this example, countriesWithLowPopulation will contain an array of objects representing countries with a population of less than 2 lakhs.

Make sure to replace the sample data with the actual data from the rest countries API or any other data source you are using. Additionally, you may need to adjust the population threshold according to the specific unit or format used in your dataset if it's different from the example provided.

**c) Print the following details name, capital, flag using forEach function**

To print the details of name, capital, and flag for each country in a dataset of countries using the forEach function, you can follow these steps:

Assuming you have an array of country objects with the properties "name," "capital," and "flag," you can use the forEach function to iterate over each country and print its details.

Here's an example code snippet:

// Sample array of country objects

const countries = [

{ name: "India", capital: "New Delhi", flag: "🇮🇳" },

{ name: "China", capital: "Beijing", flag: "🇨🇳" },

{ name: "Japan", capital: "Tokyo", flag: "🇯🇵" },

{ name: "France", capital: "Paris", flag: "🇫🇷" },

// Add more country objects here...

];

// Use the forEach function to print details for each country

countries.forEach(country => {

console.log("Name: " + country.name);

console.log("Capital: " + country.capital);

console.log("Flag: " + country.flag);

console.log(""); // Add an empty line for separation

});

In this code, the forEach function iterates over each country object in the countries array, and for each country, it prints its name, capital, and flag. The console.log("") line is used to add an empty line between each country's details for better readability.

Make sure to replace the sample data with the actual data from your dataset or API if you're using real-world data.

**d) Print the total population of countries using reduce function**

To print the total population of countries using the reduce function, you can follow these steps:

Assuming you have an array of country objects with the "population" property, you can use the reduce function to calculate the total population by iteratively summing up the populations of all countries.

Here's an example code snippet:

// Sample array of country objects with population data

const countries = [

{ name: "India", population: 1380000000 }, // 1.38 billion

{ name: "China", population: 1444216107 }, // 1.44 billion

{ name: "Japan", population: 126010000 }, // 126 million

{ name: "France", population: 67293000 }, // 67.29 million

// Add more country objects here...

];

// Use the reduce function to calculate the total population

const totalPopulation = countries.reduce((accumulator, country) => {

return accumulator + country.population;

}, 0);

console.log("Total Population: " + totalPopulation);

In this code, the reduce function iterates over each country object in the countries array and accumulates the population of each country into the accumulator variable, starting with an initial value of 0. After processing all countries, it prints the total population.

Make sure to replace the sample data with the actual data from your dataset or API if you're using real-world data.

**e) Print the country which uses US Dollars as currency.**

To print the country/countries that use US Dollars (USD) as their currency, you can use the filter function to filter out the countries that have USD as their currency code.

Here's an example code snippet:

// Sample array of country objects with currency data

const countries = [

{ name: "United States", currencies: [{ code: "USD", name: "United States Dollar" }] },

{ name: "Canada", currencies: [{ code: "CAD", name: "Canadian Dollar" }] },

{ name: "Mexico", currencies: [{ code: "MXN", name: "Mexican Peso" }] },

{ name: "Panama", currencies: [{ code: "USD", name: "United States Dollar" }] },

// Add more country objects here...

];

// Use the filter function to get countries that use USD as currency

const countriesUsingUSD = countries.filter(country => {

return country.currencies.some(currency => currency.code === "USD");

});

// Print the countries using USD as currency

countriesUsingUSD.forEach(country => {

console.log(country.name);

});

In this code, we use the filter function to filter out countries that have at least one currency with the code "USD." The some function is used to check if any of the currencies for a country have the code "USD." If so, the country is included in the countriesUsingUSD array.

Make sure to replace the sample data with the actual data from your dataset or API if you're using real-world data.