Name :Adarsh Mishra

Contact :8957331416,

Email:adarsh7268@gmail.com

1. The Bitcoin rate is available as a public REST API that gives response in json format here: https://api.coindesk.com/v1/bpi/currentprice.json Write a program that uses this REST API to get the current rate of bitcoin and prints it in words. You can ignore the decimal part.
for example: if the rate is as follows:
"rate":"22,616.3987"
Your program should print: Twenty Two Thousand Six Hundred and Sixteen.

```
package hello;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.*;

//import org.json.simple.JSONArray;

//import org.json.simple.JSONObject;

//import org.json.simple.parser.JSONParser;



public class Main {


	public static void main(String[] args) {

		ArrayList<ArrayList<String>> rate = getBitcoinRate();

		for(int i = 0 ; i < rate.get(0).size() ; i++) {

			String x = convertToWords(Integer.parseInt(rate.get(0).get(i)));

//			System.out.println(x);

			System.out.println(rate.get(1).get(i) +" : "+ x);

		}
```

```
    }

    public static ArrayList<ArrayList<String>> getBitcoinRate() {
        String apiUrl = "https://api.coindesk.com/v1/bpi/currentprice.json";
        StringBuilder response = new StringBuilder();
        ArrayList<ArrayList<String>> output = new ArrayList<>();
        output.add(new ArrayList<String>());
        output.add(new ArrayList<String>());
        try {
            URL url = new URL(apiUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            String line;
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        String json = response.toString();
        String[] str = json.split("rate_float");

        for(int i = 1 ; i < str.length ; i++) {
            String[] rate = str[i].split("\\.");
//              System.out.println(rate[0].substring(2));
            output.get(0).add(rate[0].substring(2));
```

```
            }
            for(int i = 0 ; i < str.length-1 ; i++) {
                String[] code = str[i].split("code");
//                  System.out.println();
                output.get(1).add(code[1].substring(3,6));
            }

            return output;
        }
    public static String convertToWords(int n)
        {
            long limit = 1000000000000L, curr_hun, t = 0;

            // If zero return zero
            if (n == 0)
               return ("Zero");

            // Array to store the powers of 10
            String multiplier[] = { "", "Trillion", "Billion",
                                    "Million", "Thousand" };

            // Array to store numbers till 20
            String first_twenty[] = {
               "",      "One",     "Two",     "Three",
               "Four",   "Five",    "Six",     "Seven",
               "Eight",  "Nine",    "Ten",     "Eleven",
               "Twelve",  "Thirteen",  "Fourteen", "Fifteen",
               "Sixteen", "Seventeen", "Eighteen", "Nineteen"
            };

            // Array to store multiples of ten
```

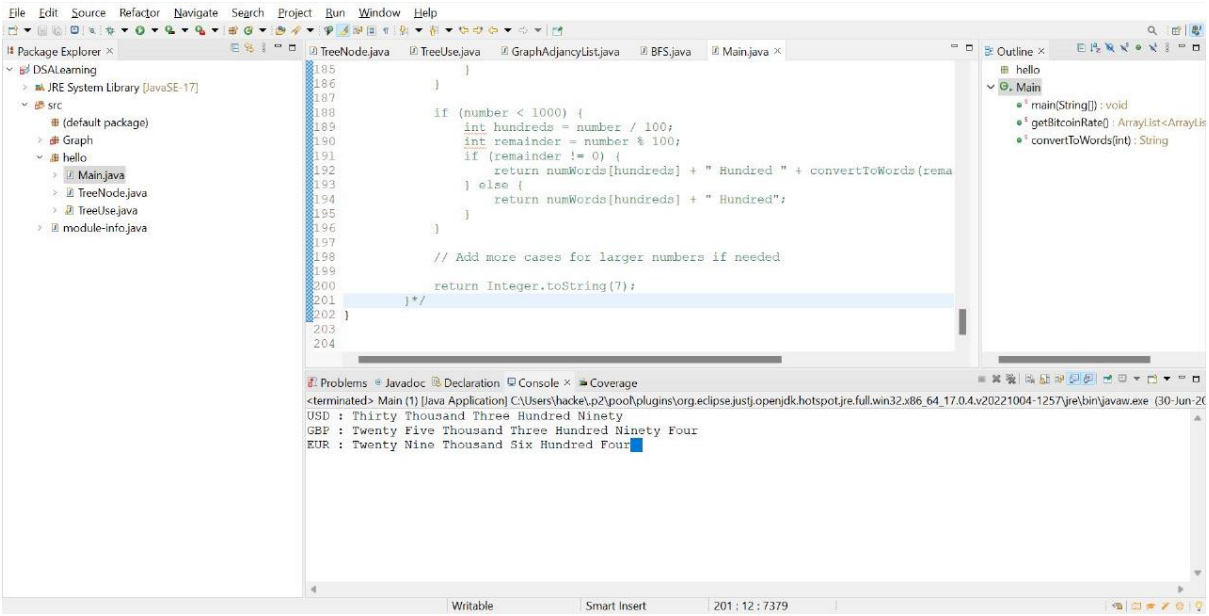
```
String tens[] = { "",       "Twenty", "Thirty",
                  "Forty",   "Fifty",  "Sixty",
                  "Seventy", "Eighty", "Ninety" };

// If number is less than 20, return without any
// further computation
if (n < 20L)
   return (first_twenty[(int)n]);
String answer = "";
for (long i = n; i > 0; i %= limit, limit /= 1000) {

   // Store the value in multiplier[t], i.e n =
   // 1000000, then r = 1, for multiplier(million),
   // 0 for multipliers(trillion and billion)
   // multiplier here refers to the current
   // accessible limit
   curr_hun = i / limit;

   // It might be possible that the current
   // multiplier is bigger than your number
   while (curr_hun == 0) {

      // Set i as the remainder obtained when n
      // was divided by the limit
      i %= limit;

      // Divide the limit by 1000, shifts the
      // multiplier
      limit /= 1000;

      // Get the current value in hundreds, as
```

```
    // English system works in hundreds
    curr_hun = i / limit;

    // Shift the multiplier
    ++t;
}

// If current hundred is greater than 99, Add
// the hundreds' place
if (curr_hun > 99)
    answer += (first_twenty[(int)curr_hun / 100]
            + " Hundred ");

// Bring the current hundred to tens
curr_hun = curr_hun % 100;

// If the value in tens belongs to [1,19], add
// using the first_twenty
if (curr_hun > 0 && curr_hun < 20)
    answer
        += (first_twenty[(int)curr_hun] + " ");

// If curr_hun is now a multiple of 10, but not
// 0 Add the tens' value using the tens array
else if (curr_hun % 10 == 0 && curr_hun != 0)
    answer
        += (tens[(int)curr_hun / 10 - 1] + " ");

// If the value belongs to [21,99], excluding
// the multiples of 10 Get the ten's place and
// one's place, and print using the first_twenty
```

```
        // array
        else if (curr_hun > 20 && curr_hun < 100)
            answer
                += (tens[(int)curr_hun / 10 - 1] + " "
                    + first_twenty[(int)curr_hun % 10]
                    + " ");

        // If Multiplier has not become less than 1000,
        // shift it
        if (t < 4)
            answer += (multiplier[(int)++t] + " ");
    }
    return (answer);
  }


/* public static String convertToWords(int number) {
    // Dictionary to map numbers to their word representations
    String[] numWords = {
        "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten",
        "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen",
        "Eighteen", "Nineteen", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy",
        "Eighty", "Ninety"
    };

    if (number == 0) {
        return numWords[0];
    }

    if (number < 20) {
        return numWords[number];
```

```
        }

        if (number < 100) {
            int tens = number / 10 * 10;
            int units = number % 10;
            if (units != 0) {
                return numWords[tens] + " " + numWords[units];
            } else {
                return numWords[tens];
            }
        }

        if (number < 1000) {
            int hundreds = number / 100;
            int remainder = number % 100;
            if (remainder != 0) {
                return numWords[hundreds] + " Hundred " + convertToWords(remainder);
            } else {
                return numWords[hundreds] + " Hundred";
            }
        }

        // Add more cases for larger numbers if needed

        return Integer.toString(7);
    }*/
}
```

Output



Q2 Given a matrix of characters representing a place on Earth, where the value 'T' indicates the

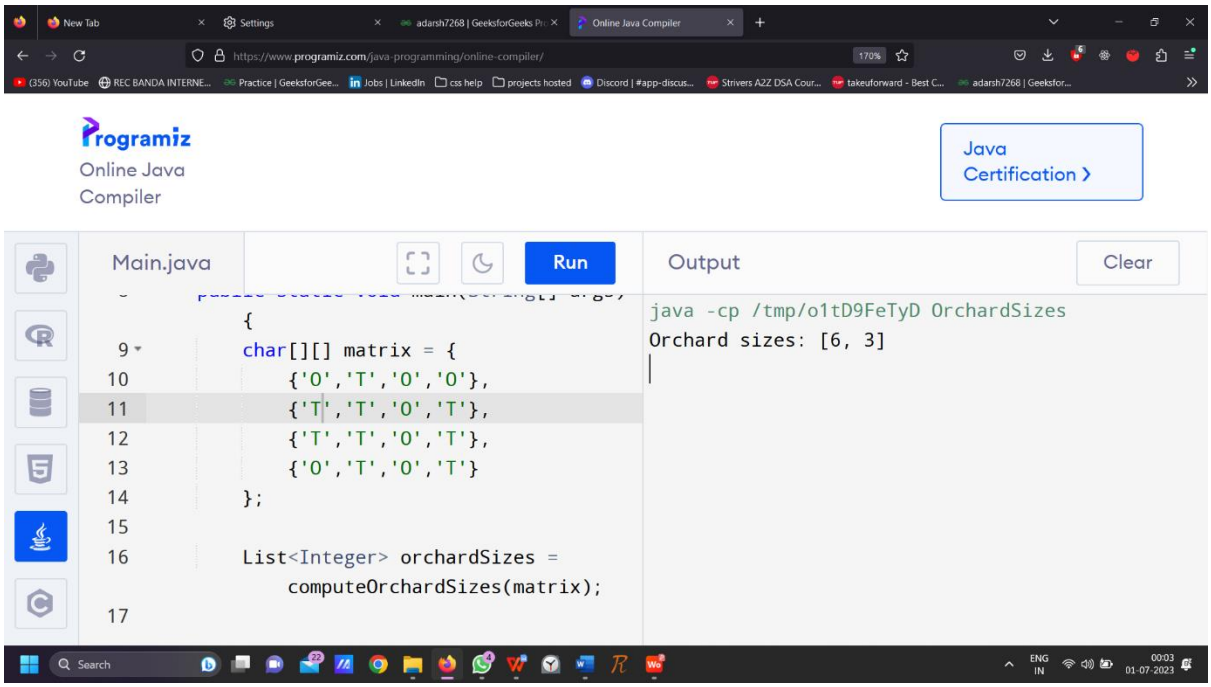
presence of a Tree at that location and 'O' represents there is no tree at that point. An orchard is a region with tress connected vertically, horizontally, or diagonally. The size of the orchard is the total number of connected trees. Write a method to compute the sizes of all orchards in the matrix.
Example input:
[
['O','T','O','O'],
['O','T','O','T'],
['T','T','O','T'],
['O','T','O','T']
]
Note: Input can be in the code itself (it doesn't have to be supplied at runtime)
Output: 5, 3

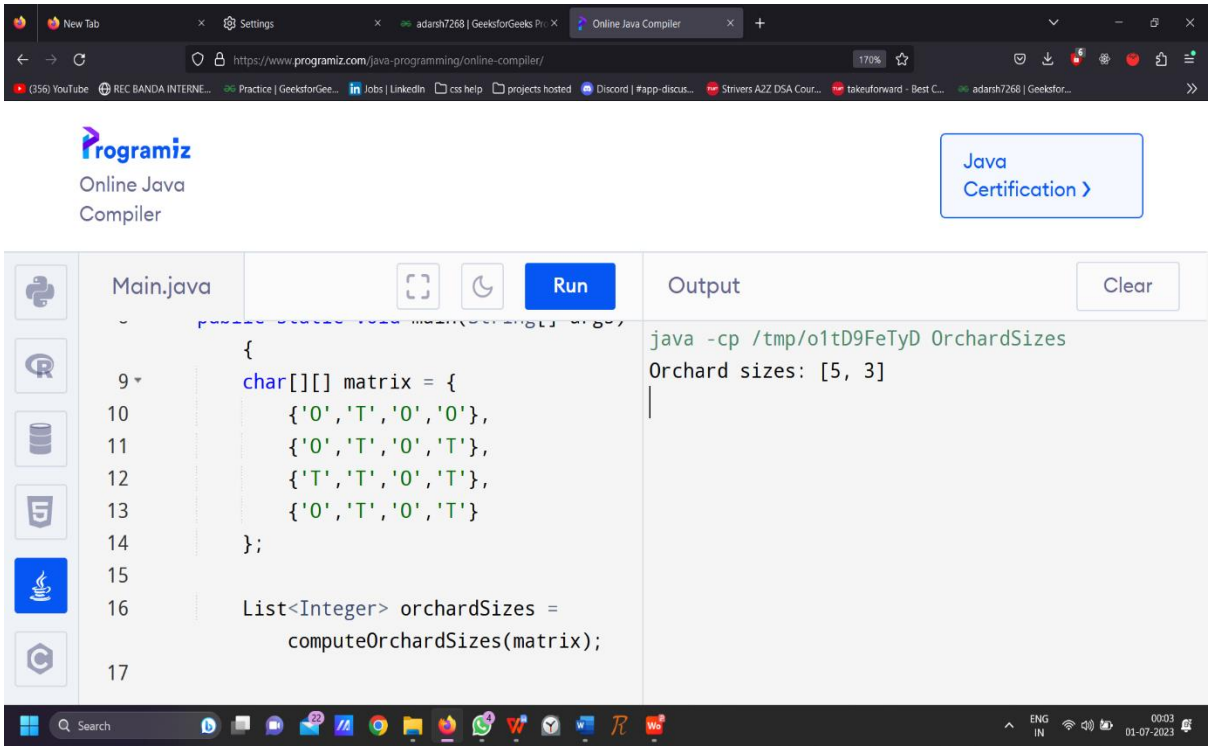
```
import java.util.ArrayList;

import java.util.List;


public class OrchardSizes {

    public static void main(String[] args) {

        char[][] matrix = {

            {'O','T','O','O'},

            {'O','T','O','T'},

            {'T','T','O','T'},

            {'O','T','O','T'}

        };


        List<Integer> orchardSizes = computeOrchardSizes(matrix);


        System.out.println("Orchard sizes: " + orchardSizes);

    }


    public static List<Integer> computeOrchardSizes(char[][] matrix) {

        List<Integer> orchardSizes = new ArrayList<>();

        int rows = matrix.length;

        int cols = matrix[0].length;
```

```
    boolean[][] visited = new boolean[rows][cols];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] == 'T' && !visited[i][j]) {
                int orchardSize = findOrchardSize(matrix, visited, i, j);
                orchardSizes.add(orchardSize);
            }
        }
    }

    return orchardSizes;
}

private static int findOrchardSize(char[][] matrix, boolean[][] visited, int row, int col) {
    int size = 0;
    int rows = matrix.length;
    int cols = matrix[0].length;

    if (row < 0 || row >= rows || col < 0 || col >= cols || visited[row][col] || matrix[row][col] != 'T') {
        return size;
    }

    visited[row][col] = true;
    size++;

    size += findOrchardSize(matrix, visited, row - 1, col); // Check up
    size += findOrchardSize(matrix, visited, row + 1, col); // Check down
    size += findOrchardSize(matrix, visited, row, col - 1); // Check left
    size += findOrchardSize(matrix, visited, row, col + 1); // Check right
    size += findOrchardSize(matrix, visited, row - 1, col - 1); // Check diagonally up-left
```

```
        size += findOrchardSize(matrix, visited, row - 1, col + 1); // Check diagonally up-right
        size += findOrchardSize(matrix, visited, row + 1, col - 1); // Check diagonally down-left
        size += findOrchardSize(matrix, visited, row + 1, col + 1); // Check diagonally down-right

        return size;
    }
}
```

Output




New Tab
Settings
adarsh7268 | GeeksforGeeks
Online Java Compiler
https://www.programiz.com/java-programming/online-compiler/
Programiz
Online Java
Compiler
Java
Certification
Main.java
Run
Output
Clear
{
char[][] matrix = {
{'O','T','O','O'},
{'O','T','O','T'},
{'T','T','O','T'},
{'O','T','O','T'}
};
List<Integer> orchardSizes =
computeOrchardSizes(matrix);
java -cp /tmp/o1tD9FeTyD OrchardSizes
Orchard sizes: [5, 3]
Search
ENG
IN
00:03
01-07-2023