МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №5**

з дисципліни «Веб-технології та веб-дизайн-1»

**Виконав:**

студент гр. БС-83

Васильєв А.А

**Перевірив:**

ас. каф. БМК Матвійчук О. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання**

Завдання:

В результаті у кожного мають бути функції для виконання 3х завдань,

1 - обов’язкове для всіх,

2 - variant mod 6

3 - (((group mod 10) + variant) mod 6) + 6

Завдання обов'язкове для всіх:

Фіскальний кодекс

Кожна особа в Італії має унікальний ідентифікаційний ідентифікаційний код, виданий національною податковою службою після реєстрації народження: Фіскальний кодекс (Codice Fiscale). [https://en.wikipedia.org/wiki/Italian\_fiscal\_code\_card]

Враховуючи об'єкт, що містить особисті дані людини (ім'я, прізвище, стать та дата народження), поверніть 11 символів коду як рядок, виконуючи наступні кроки:

Створіть 3 символи прізвища, якщо воно має:

Принаймні 3 приголосні, тоді використовуються перші три приголосні. (Ньюмен -> NWM).

Менше ніж 3 приголосні, тоді голосні замінять відсутні літери в тому ж порядку, в якому вони відображаються (Fox -> FXO | Hope -> HPO).

Менше трьох літер тоді "X" займе третю позицію після приголосного та голосного (Yu -> YUX).

Створіть від імені 3 великі літери, якщо у них є:

Рівно 3 приголосних тоді приголосні вживаються в тому порядку, в якому вони з’являються (Метт -> МТТ).

Вживається більше 3 приголосних, тоді перший, третій та четвертий приголосні (Саманта -> СНТ | Томас -> ТМС).

Менше трьох приголосних, тоді голосні замінять відсутні літери в тому ж порядку, в якому вони відображаються (Bob -> BBO | Paula -> PLA).

Менше трьох літер тоді "X" займе третійю позицію після приголосного та голосного (Al -> LAX).

Утворіть 2 числа, 1 букву та 2 числа від дати народження та статі:

Візьміть дві останні цифри року народження (1985 -> 85).

Створіть лист, відповідний місяцю народження (січень -> A | грудень -> T), використовуючи таблицю для перерахунку, включену в код.

Для чоловіків приймають день народження, додаючи один нуль на початку, якщо він менше 10 (будь-який 9-й день -> 09 | будь-який 20-й день -> 20).

Для жінок беруть день народження і сумують до нього 40 (будь-який 9-й день -> 49 | будь-який 20-й день -> 60).

PS:

· Букви в відповіді мають бути в uppercase

· Дата народження в форматі D/M/YYYY

· Таблиця перетворення наведена в завданні

· Для людини створити клас

· Відображення місяця в літеру: // const months = { 1: "A", 2: "B", 3: "C", 4: "D", 5: "E", 6: "H", 7: "L", 8: "M", 9: "P", 10: "R", 11: "S", 12: "T" }

2. Відстань до найближчого голосного

Напишіть функцію, яка займає рядок і для кожного символу повертає відстань до найближчого голосного. Якщо символ сам є голосним, поверніть 0.

мова роботи функції англійська, тому голосними є “a, e, i, o, u”, також всі букви будуть в lowercase.

Створіть функцію, яка займає масив і повертає елемент який зустрічається найчастіше.

Приклад:

findFrequent([3, 7, 3]) ➞ 3

findFrequent([null, "hello", true, null]) ➞ null

findFrequent([false, "up", "down", "left", "right", true, false]) ➞ false

**Лістинг**

**code\_fiscale.js**

/\*\* class which represens a person\*/

class person{

  constructor(name, surname, gender, dateOfBirth){

  this.name=name;

  this.surname=surname;

  this.gender=gender;

  this.dateOfBirth=dateOfBirth;

  }

/\*\*getters\*/

  get Name(){

    return this.name\_;

  }

  get Surname(){

    return this.surname\_;

  }

  get Gender(){

    return this.gender\_;

  }

  get DateOfBirth(){

    return this.dateOfBirth\_;

  }

}

/\*\*two people from the example\*/

let Helen = new person("Helen","Yu","F","1/12/1950");

var Matt = new person("Matt","Edabit","M","1/1/1900");

/\*\*actually function\*/

function fiscaleCodeGenerator(certain\_person){

  /\*\*whole code consists of parts made from different person parameters\*/

   var nameCode="", surnameCode="", dobCode="";

   var allConsonants=[

     "B","C","D","F","G",

     "H","J","K","L","M",

     "N","P","Q","R","S",

     "T","V","W","X","Z"];

   var name=certain\_person.Name.toUpperCase();

   var surname=certain\_person.Surname.toUpperCase();

   var gender=certain\_person.Gender.toUpperCase();

   var dob=certain\_person.DateOfBirth;

   /\*\*computing surname code part\*/

   if(surname.length<3){

     surnameCode=surname+"X";

   } else{

     consonants="";

     vowels="";

     for(char of surname){

       for(i=0; i<19; i++){

         if(char===allConsonants[i]){

           consonants=consonants+char;

         } else{

           vowels=vowels+char;

         }

       }

    }

    surnameCode=(consonants+vowels).slice(0,3);

  }

  /\*\*computing name code part\*/

  if(name.length<3){

    nameCode=name+"X";

  } else{

    consonants="";

    vowels="";

    var counter=0;

    for(char of surname){

      for(i=0; i<19; i++){

        if(char===allConsonants[i]){

          consonants=consonants+char;

          counter++;

        } else{

          vowels=vowels+char;

        }

      }

   }

   if(counter<=3){

     nameCode=(consonants+vowels).slice(0,3);

   } else if(counter==3){

     nameCode=consonants;

   } else if(counter>=3){

     nameCode=consonants[0]+consonants[2]+consonants[3];

   }

 }

 /\*\*computing date of birth code part\*/

 const monthsTable = {

   1: "A", 2: "B", 3: "C",

   4: "D", 5: "E", 6: "H",

   7: "L", 8: "M", 9: "P",

   10: "R", 11: "S", 12: "T"};

  /\*\* computing year\*/

  var yearCode=dob.substr(6,2);

  /\*\* computing day\*/

  var slashIndex=0;

  isDoubleDigit=false;

  var day=dob.substr(0,3);

  if (day[2]=="/"){

    slashIndex=2;

    isDoubleDigit=true;

    day=day[0]+day[1];

  } else{

    day=day[0];

    slashIndex=1;

  }

  if(gender=="M"&&isDoubleDigit==false){

      day="0"+day;

  } else if(gender=="M"){

    day=Number(day);

    day=day+40;

    day=String(day);

  }

  var dayCode=day;

  /\*\*computing month\*/

  var month=dob.substr(slashIndex+1,2);

  if(month[1]=="/"){

    month=month[0];

  }

  var monthCode=monthsTable[Number(month)];

  /\*\*Compiling patrs\*/

  var fiscaleCode=surnameCode+nameCode+yearCode+monthCode+dayCode;

  return fiscaleCode;

}

console.log(fiscaleCodeGenerator(Helen))

**most\_frequent\_element.js**

function mostFrequentElementCounter(array){

  var elementsEntries=[]

  var arrayLength=array.length

  for(i=0;i<arrayLength; i++){

    elementsEntries.push([array[i], 1])

    var currentMostFrequentIndex=0

    var currentmostFrequentElement=array[currentMostFrequentIndex]

    for(j=0;j<elementsEntries.length;j++)

    {

      if(i!=j && array[i]==elementsEntries[j][0]){

        elementsEntries[j][1]++

      }

      if(elementsEntries[currentMostFrequentIndex][1]<elementsEntries[j][1]){

        currentMostFrequentIndex=j

      }

    }

  }

  console.log(`Most frequent element ${array[currentMostFrequentIndex]}, index ${currentMostFrequentIndex}`)

  return [array[currentMostFrequentIndex], currentMostFrequentIndex]

}

mostFrequentElementCounter([1,1,3,1,4,5,4,5,4,3,4])

**vowels\_distance.js**

function shortestDistancesToAllVowels(string){

  /\*service variables\*/

  var wordLength=string.length

  var vowelsTable=["a","o","u","e","i"], vowelsIndexes=[]

  /\*finding vowels\*/

  for(i=0;i<wordLength;i++){

    for(letter of vowelsTable){

      if(letter==string[i]){

        vowelsIndexes.push(i)

      }

    }

  }

  /\*finding distances\*/

  var shortestDistancesTable=[]

  var vowelCount=vowelsIndexes.length

  var distancesToAllVowels

  for(i=0; i<wordLength;i++){

    let distancesToAllVowels=[]

    for(j=0;j<vowelCount;j++){

      distancesToAllVowels.push(Math.abs(i-vowelsIndexes[j]))

    }

    shortestDistancesTable.push(Math.min.apply(null,distancesToAllVowels))

  }

  return shortestDistancesTable

}

console.log(shortestDistancesToAllVowels("hopper"))

console.log(shortestDistancesToAllVowels("Nebuchadnezzar"))