 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»

**Виконала:**

студентка гр. БС-81

Соловйова А.О.

**Перевірив:**

ас. каф. БМК Матвійчук О. В.

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

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

(підпис викладача)

Київ-2020

**Завдання:**

1. **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" }

1. **Візерунчастий ремінець**

Браслет може мати 4 візерунка:

* горизонтальний: кожен елемент підряд однаковий.
* вертикальний: кожен елемент у стовпці однаковий.
* діагональна зліва: кожен елемент ідентичний тому, що знаходиться вгорі ліворуч або праворуч знизу.
* діагональна справа: кожен елемент ідентичний тому, що знаходиться вгорі праворуч або внизу зліва.

Напишіть функцію, яка повертає True, якщо розділ можна правильно класифікувати на один із 4 типів, в іншому випадку False.

*Приклад:*

isWristband([

  ["A", "A"],

  ["B", "B"],

  ["C", "C"]

]) ➞ true

// Part of horizontal wristband.

isWristband([

  ["A", "B"],

  ["A", "B"],

  ["A", "B"]

]) ➞ true

// Part of vertical wristband.

isWristband([

  ["A", "B", "C"],

  ["C", "A", "B"],

  ["B", "C", "A"],

  ["A", "B", "C"]

]) ➞ true

// Part of diagonal left wristband.

isWristband([

  ["A", "B", "C"],

  ["B", "C", "A"],

  ["C", "A", "B"],

  ["A", "B", "A"]

]) ➞ true

// Part of diagonal right wristband.

1. **Видаліть останню голосну**

Напишіть функцію, яка видаляє останню голосну в кожному слові речення.

Функція повинна працювати з англійським алфавітом.

Голосні - a, e, i, o, u

Всі букви будуть в lowercase

*Приклад:*

removeLastVowel("Those who dare to fail miserably can achieve greatly.")

➞ "Thos wh dar t fal miserbly cn achiev gretly."

removeLastVowel("Love is a serious mental disease.")

➞ "Lov s  serios mentl diseas"

removeLastVowel("Get busy living or get busy dying.")

➞ "Gt bsy livng r gt bsy dyng"

**Лістинг:**

**Lab5.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Lab5</title>

</head>

<body>

<center>

<h1>LAB 5</h1>

<h2><b>Soloviova Anastasiia</b></h2>

<script src="task1.js"></script>

<script src="task2.js"></script>

<script src="task3.js"></script>

</center>

</body>

</html>

**task1.js**

class Person

{

constructor(name,surname,bd,gender)

{

this.Name=name;

this.Surname=surname;

this.BirthDay=bd;

this.Gernder=gender;

}

fiscal\_code() {

return (this.Surname\_code()+this.Name\_code() + this.BD\_code() ).toUpperCase();

}

vowel(letter) {

return ['a', 'e', 'i', 'o', 'u'].indexOf(letter.toLowerCase()) !== -1;

}

get\_bd(olddate)

{

let new\_bd = olddate.split('/').map(function(val)

{

return parseInt(val, 10);

})

return new\_bd;

}

BD\_code() {

const mounth = { 1: "A", 2: "B", 3: "C", 4: "D", 5: "E", 6: "H", 7: "L", 8: "M", 9: "P", 10: "R", 11: "S", 12: "T" }

let data = this.get\_bd(this.BirthDay);

let year = data[2] %100;

if(year<10){

year= '0'+ year;

}

let day= this.Gernder== 'F'? data[0]+ 40: data[0];

if(day<10){

day='0'+day;

}

let BD\_code = year+ mounth[data[1]] + day;

return BD\_code;

}

Name\_code() {

return this.\_codeFromNames(this.\_delSecondCons(this.Name));

}

\_delSecondCons(string)

{

let consIdx = [];

for (let i = 0; i < string.length; i++)

{

if (!this.vowel(string[i])) {

consIdx.push(i);

}

}

if (consIdx.length > 3) {

return string.slice(0, consIdx[1]) + string.slice(consIdx[1]+1, string.length)

}

return string;

}

Surname\_code() {

return this.\_codeFromNames(this.Surname);

}

\_codeFromNames(string)

{

let constonants = '';

let vowels = '';

for (let i = 0; i < string.length; i++){

if (this.vowel(string[i])) {

vowels+= string[i];

}

else {

constonants+=string[i];

}

}

let code = constonants + vowels;

if (code.length>=3){

return code.slice(0, 3);

}

return code+"X".repeat(3-code.length);

}

}

NewPerson = new Person('Matt',"Edabit","1/1/1900",'M')

console.log("Task 1");

console.log(NewPerson.fiscal\_code())

**task2.js**

var array1 =

[

["A", "B"],

["A", "B"],

["A", "B"]

];

var array2 =

[

["A","B","C"],

["B","C","A"],

["C","A","B"],

["C","B","C"]

];

function isWristBand(arr)

{

// row

var val1 = true, val2 = true, val3 = true, val4 = true;

for(i = 0; i < arr.length; ++i)

{

verification = true;

for(j = 1; j < arr[i].length; ++j)

{

if(arr[i][j]!==arr[i][j-1])

{

verification = false;

}

}

if(verification === false)

{

val1 = false;

}

}

// column

for(j = 0; j < arr[0].length; ++j)

{

verification = true;

for(i = 1; i < arr.length; ++i)

{

if(arr[i][j] !== arr[i-1][j])

{

verification = false;

}

}

if(verification === false)

{

val2 = false;

}

}

// diagonal left

for(i = 0; i < arr.length; ++i)

{

for(j = 0; j < arr[0].length; ++j)

{

temp = "";

for(z = 0; z < arr[0].length-j && z < arr.length-i; ++z)

{

temp+= arr[i+z][j+z];

}

if((temp.split(temp[0]).length - 1)!== temp.length)

{

val3=false;

}

}

}

// diagonal right

for(i = 0; i < arr.length; ++i)

{

for(j = arr[0].length-1; j >= 0 ; --j)

{

temp = "";

for(z = 0; z < arr.length-i && j-z>=0; ++z)

{

temp+= arr[i+z][j-z];

}

if((temp.split(temp[0]).length - 1)!== temp.length)

{

val4=false;

}

}

}

return val1||val2||val3||val4;

}

console.log("Task 2");

console.log(array1[0][0] + " " + array1[0][1]);

console.log(array1[1][0] + " " + array1[1][1]);

console.log(array1[2][0] + " " + array1[2][1]);

console.log("Result:" + isWristBand(array1));

console.log(array2[0][0] + " " + array2[0][1] + " " + array2[0][2]);

console.log(array2[1][0] + " " + array2[1][1] + " " + array2[1][2]);

console.log(array2[2][0] + " " + array2[2][1] + " " + array2[2][2]);

console.log(array2[3][0] + " " + array2[3][1] + " " + array2[3][2]);

console.log("Result:" + isWristBand(array2));

**task3.js**

function removeLastVowel(arr) {

const vowels = ['a', 'e', 'i', 'o', 'u']

let ans = arr.split(" ");

for (i = 0; i < ans.length; i++)

{

for (j = ans[i].length - 1; j >= 0; j--)

{

if (vowels.includes(ans[i][j]))

{

ans[i] = ans[i].slice(0, j) + ans[i].slice(j + 1);

break;

}

}

}

return ans.join(' ');

}

console.log("Task 3");

console.log(removeLastVowel("Those who dare to fail miserably can achieve greatly."));