

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.»

Варіант №15

**Виконала:**

студентка гр. БС-83

Стегніщева О. М.

**Перевірив:**

ас. Давидько О. Б.,

ас. Матвійчук О. В.

Зараховано від

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

(підпис викладача)

Київ-2020

**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" }

*Приклад:*

fiscalCode({

name: "Matt",

surname: "Edabit",

gender: "M",

dob: "1/1/1900"

}) ➞ "DBTMTT00A01"

fiscalCode({

name: "Helen",

surname: "Yu",

gender: "F",

dob: "1/12/1950"

}) ➞ "YUXHLN50T41"

**Завдання по варіантах:**

**3 Виткані струни**

Створіть функцію, яка повертає масив заданого рядка, але зміщений пробілами. Ось кілька більш точних інструкцій:

Продовжуйте додавати пробіли зліва, поки у вас не буде однакової кількості пробілів, як довжина слова.

Потім продовжуйте видаляти пробіли, поки не досягнете початкового слова.

*Приклад:*

wiggleString("hello") ➞ [

"hello",

 " hello",

 "  hello",

 "   hello",

 "    hello",

 "     hello"

 "    hello",

 "   hello",

 "  hello",

 " hello",

 "hello"

 ]

**6 Сполучні слова**

Напишіть функцію, яка з'єднує кожне попереднє слово з наступним словом спільними літерами. Повертайте отриманий рядок (видаляючи дублікати символів у накладенні) та мінімальну кількість спільних літер у всіх парах рядків.

*Приклад:*

join(["oven", "envier", "erase", "serious"]) ➞ ["ovenvieraserious", 2]

join(["move", "over", "very"]) ➞ ["movery", 3]

join(["to", "ops", "psy", "syllable"]) ➞ ["topsyllable", 1]

// "to" and "ops" share "o" (1)

// "ops" and "psy" share "ps" (2)

// "psy" and "syllable" share "sy" (2)

// the minimum overlap is 1

//index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>index</title>

</head>

<body>

<script src='1.js'></script>

<script src='2.js'></script>

<script src='3.js'></script>

</body>

</html>

//1.js

console.log(' TASK 1: Italian fiscal code ');

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" }

class Human {

constructor(name, surname, gender, dob) {

this.name = name;

this.surname = surname;

this.gender = gender;

this.dob = dob;

}

}

var x = new Human('Aleksandra', 'Stegnishcheva', 'female', '5.6.2001');

console.log(x);

var y = new Human('Matt', 'Edabit', 'male', '1.1.1900');

console.log(y);

var z = new Human('Helen', 'Yu', 'female', '1.12.1950');

console.log(z);

function removeVowels(word) {

return word.replace(/[aeiouy]/gi, '');

}

function removeConsonant(word) {

return word.replace(/[bcdfghjklmnpqrstvwxz]/gi, '');

}

function surname\_f(word) {

var new\_surname = '';

if (removeVowels(word).length >= 3) {

for (var i = 0; i < 3; i++) {

new\_surname += removeVowels(word)[i];

}

return new\_surname.toUpperCase();

}

if (removeVowels(word).length < 3 && word.length >= 3) {

new\_surname += removeVowels(word);

for (i = 0; i < 3; i++) {

if (new\_surname.length != 3) {

new\_surname += removeConsonant(word)[i];

} else {

new\_surname += 'X';

}

}

return new\_surname.toUpperCase();

}

if (word.length < 3) {

new\_surname += removeVowels(word) + removeConsonant(word) + 'X';

return new\_surname.toUpperCase();

}

}

function name\_f(word) {

var new\_name = '';

if (removeVowels(word).length == 3) {

new\_name = removeVowels(word);

return new\_name.toUpperCase();

}

if (removeVowels(word).length > 3) {

for (var i = 0; i < 4; i++) {

if (i == 1) {

continue;

}

new\_name += removeVowels(word)[i];

}

return new\_name.toUpperCase();

}

if (removeVowels(word).length < 3 && word.length >= 3) {

new\_name += removeVowels(word);

for (i = 0; i < 3; i++) {

if (new\_name.length != 3) {

new\_name += removeConsonant(word)[i];

} else {

break;

}

}

return new\_name.toUpperCase();

}

if (word.length < 3) {

new\_name += removeVowels(word) + removeConsonant(word) + 'X';

return new\_name.toUpperCase();

}

}

function data\_gender(dob, gen) {

var data = '';

var res = '';

for (i = 0; i < 2; i++) {

data = dob.split('.');

}

day = data[0];

month = data[1];

year = data[2];

res = year.substr(-2);

res += months[month];

if (gen == 'male') {

if (day < 10) {

res += '0' + day;

} else {

res += day;

}

} else {

var newday = +day;

newday += 40;

res += newday;

}

return (res);

}

console.log(surname\_f(x.surname) + name\_f(x.name) + data\_gender(x.dob, x.gender));

console.log(surname\_f(y.surname) + name\_f(y.name) + data\_gender(y.dob, y.gender));

console.log(surname\_f(z.surname) + name\_f(z.name) + data\_gender(z.dob, z.gender));

//2.js

console.log(' ');

console.log(' TASK 2: Wiggle string ');

function wiggleString(word) {

var arr = [];

for (var i = 0; i <= word.length; i++){

arr.push(' '.repeat(i) + word);

}

for (var i = word.length - 1; i >= 0; i--){

arr.push(' '.repeat(i) + word);

}

return arr;

}

console.log(wiggleString('hello'));

//3.js

console.log(' ');

console.log(' TASK 3: Join words ');

var a = ["oven", "envier", "erase", "serious"];

var b = ["move", "over", "very"];

var c = ["to", "ops", "psy", "syllable"]

function join(arr) {

var newWords = arr[0];

var overlaps = 345356;

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

var l = 1;

while (!arr[i-1].endsWith(arr[i].slice(0, l))){

l += 1;

}

if (l < overlaps){

overlaps = l;

}

newWords = newWords.slice(0, -l);

newWords += arr[i];

}

return [newWords, overlaps];

}

console.log(join(a));

console.log(join(b));

console.log(join(c));

**Результати роботи:**





