 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

Розмариця О. А.

**Перевірив:**

ас. каф. БМК Давидько О. Б.

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" }

*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:**

1. **Числа харшад**

Числа Харшада / Нівена - це додатні числа, які поділяються на суму їх цифр. Усі одноцифрові числа - це харшадські номери.

Наприклад, 27 - це число Харшада, як 2 + 7 = 9, а 9 - дільник 27.

Різні номери можуть виникати в послідовних кластерах. Числа від 1 до 10 - це харшадські числа. Числа 132 і 133 - це обидва числа харшада. Числа 1014, 1015, 1016, 1017 - це харшадські числа.

Створіть функцію, яка приймає число і повертає чи дане число є числом Харшада

*Приклад:*

harshad(5) ➞ true

harshad(82) ➞ false

// 82 mod (8 + 2) != 0

1. **Зростання послідовних чисел**

Напишіть функцію, яка повертає істину, якщо рядок складається з в**исхідних та послідовних** чисел.

ps: Число може складатися з будь-якої кількості цифр, якщо цифри прилягають одна до одної, то рядок має щонайменше дві з них.

*Приклад:*

ascending("2324256") ➞ false

// No matter how this string is divided, the numbers are not consecutive.

ascending("444445") ➞ true

// Consecutive numbers 444 and 445.

let Iname = '';

let Isurname = '';

let male = true;

let day = 0;

let month = 0;

let year = '';

class Person

{

constructor(Name, Surname, Male, Day, Month, Year)

{

this.name = Name;

this.surname = Surname;

this.Male = Male;

this.day = Day;

this.month = Month;

this.year = Year;

}

get\_fiscal\_code()

{

return (this.surname\_fiscal(this.surname) + this.name\_fiscal(this.name) + this.date\_fiscal());

}

surname\_fiscal(Surname)

{

const vowels = "eyuioaуеыаоэяиё";

Surname = Surname.toLowerCase();

let code = '';

let letters = 0;

for (let i of Surname)

{

let isVowel = false;

for (let j of vowels)

{

if (j == i)

isVowel = true;

}

if (!isVowel && letters < 3)

{

code += i;

letters++;

}

}

if (letters < 3)

{

for (let i of Surname)

{

for (let j of vowels)

{

if (j == i)

{

code += i;

letters++;

}

}

if (letters == 3)

break;

}

}

if (letters < 3)

{

for (let i = 0; i < (3 - Surname.length); i++)

code += 'x';

}

return code.toUpperCase();

}

name\_fiscal(Name)

{

const vowels = "eyuioaуеыаоэяиё";

Name = Name.toLowerCase();

let code = '';

let letters = 0;

for (let i of Name)

{

let isVowel = false;

for (let j of vowels)

{

if (j == i)

isVowel = true;

}

if (!isVowel && letters < 4)

{

code += i;

letters++;

}

}

if (letters == 4)

code = code.replace(code[1], '');

if (letters < 3)

{

for (let i of Name)

{

for (let j of vowels)

{

if (j == i)

{

code += i;

letters++;

}

}

if (letters == 3)

break;

}

}

if (letters < 3)

{

for (let i = 0; i < (3 - Name.length); i++)

code += 'x';

}

return code.toUpperCase();

}

date\_fiscal()

{

const letter\_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" }

let code = '';

code += this.year[2] + this.year[3] + letter\_months[month] + ((male) ? (this.day < 10) ? "0" + this.day : this.day : 40 + Number(this.day));

return code;

}

}

function output\_fiscal\_code()

{

let human = new Person (Iname, Isurname, male, day, month, year);

let code = document.getElementById("fiscal\_output").innerHTML = human.get\_fiscal\_code();

}

harsh = '';

function harshad()

{

let sum = 0;

let num = harsh;

while (harsh >= 1)

{

sum += harsh%10;

harsh = Math.trunc(harsh/10);

}

document.getElementById('harshad\_output').innerHTML = (num%sum == 0) ? "Yes" : "No";

}

zrostnum = 0;

function task3()

{

let ascending = false;

for (let i = 1; i <= Math.trunc(zrostnum.length/2); i++)

{

ascending = true;

let start = 0;

for (let j = i; start < zrostnum.length-j; )

{

if(Number(zrostnum.slice(start+j, start+j+j)) - Number(zrostnum.slice(start, start+j)) == 1)

start += j;

else if (Number(zrostnum.slice(start+j, start+j+j+1)) - Number(zrostnum.slice(start, start+j)) == 1)

{

start += j;

j++;

}

else

{

ascending = false;

break;

}

}

if (ascending)

break;

}

document.getElementById('task3\_output').innerHTML = ascending ? "Yes" : "No";

}