**JS Task tow**

21-March-2022

monday

Note: Create new folder “JS-21-march-2022” contains html page in your JavaScript repo.

// All the exercises should use while loop:

// Please be sure to conect this file to another HTML file

// to seprate it from the recursion file

// dDont serach on the internet except if it is written in the question

// You should solve all the question by what you understand from the lecture

// and with your pair not solo

// And dont look or use a code that you wrote before

console.log('Here is : While')

/\*

1

Create a function called subtract

that takes a single parameter n,

and return the subtraction of all

starting from n to 0

Ex: subtract(2); => 2 - 1 - 0 => 1

Ex: subtract(5); => 5 - 4 - 3 - 2 - 1 - 0 => -5

Ex: subtract(9); => 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 => -27

\*/

/\*

2

Create a function called factorial

that takes a single parameter n,

and return the product of all integers up to n starting from 1

Ex: factorial(2); => 2 \* 1 => 2

Ex: factorial(4); => 4 \* 3 \* 2 \* 1 => 24

\*/

/\*

3

Create a function called repeatStr

that takes 2 parameters string and number,

and return the string number time with space

Important: the continue condition should be [somthing !== 0]

\*\* Think how we can return nothing ???

\*\* Important: You should dint have extra space at the end

Ex: repeatStr("to",2); => "to to"

Ex: repeatStr("to",4); => "to to to to"

\*/

/\*

4

Create a function called sum2

that take two parameter

and will return the sumation from the first number to the second number

Ex: sum2(4, 5); => 4 + 5 => 9

Ex: sum2(3, 6); => 3 + 4 + 5 + 6 => 18

\*/

/\*

5

Create a function called repeatStr2

that takes 2 string parameters,

and return the first string number time

(of the lenght of the secend string) with space betweeen it

\*\* Important: the continue condition should be [somthing!== 0]

\*\* Important: You should't have extra space at the end

Ex: repeatStr2("ro","cc"); => "ro ro"

Ex: repeatStr2("ro","fff"); => "ro ro ro"

\*/

/\*

6

Create a function called multiOf

that takes 3 parameters,

and return the first number mutiple by

(the second one ^ the third number)

\*\* Important: dont use \*\*

Ex: multiOf(4,10,3); => 4000

Ex: multiOf(6,3,2); => 54

Ex: multiOf(6,2,3); => 48

\*/

/\*

7

Create a function called muti2

that take two parameter

and will return the multiplication

from the first number to the second number

Ex: muti2(4, 5); => 4 \* 5 => 20

Ex: muti2(3, 6); => 3 \* 4 \* 5 \* 6 => 360

\*/

/\*

8

Create a function called numberBetweenUs

that take 2 parameters

and return the number between them

\*\* Important: You should't have extra space at the end and comma

\*\* Important: the stop condition should [num1!==num2-1]

\*\* hint: maybe you will have 2 if statment

numberBetweenUs(2,8) => "3, 4, 5, 6, 7"

numberBetweenUs(1,3) => "2"

\*/

/\*

9

Write a function called countDown

that takes one parameter

and return a string represent the count down from the number

that insert to 0

instead of 0 you shoulf have "done"

between each number a comma and space

\*\* you need to use recursion

\*\* Important: You should't have extra space at the end and comma

countDown(5)

=> "5, 4, 3, 2, 1, done"

countDown(2)

=> "2, 1, done"

countDown(7)

=> "7, 6, 5, 4, 3, 2, 1, done"

\*/

/\*

10

Write a function called multiplication2

that takes two parameters

and return the multiplication of them

you need to use only sum

multiplication2(5,4) => 20

multiplication2(2,8) => 16

multiplication2(7,6) => 42

\*/

/\*

11

Write a function called mod2

that takes two parameters

and return the module of them

without using %

mod2(5,4) => 1

mod2(2,8) => 2

mod2(7,4) => 3

mod2(8,4) => 0

\*/

/\*

12

Write a function called repeatChar

that takes two parameters

and return the times that this char

without case sensetive

repeat inside this string

\*\*hint: "str"[0] => "s"

\*\*hint: "car"[1] => "a"

repeatChar("schOol","o") => 2

repeatChar("school","a") => 0

repeatChar("School","s") => 1

try more case by yourself

\*/

// Advanced Part {for your benefits}

// Dont solve any question here if you didnt

//finish all the question a bove with your pair

// This part you can try it and solve solo not with your pair

// This part is extra (and there is no help from the trainer) it is good to try it

/\*

1

Create a function called fibon

that takes 1 parameter a number,

and return the fibonacci for this number

Ex: fibon(7); => 21

Ex: fibon(2); => 2

Ex: fibon(1); => 1

Ex: fibon(0); => 1

fib(n)=fin(n-1)+fib(n-2)

fibon(n): 1 1 2 3 5 8 13 21

| | | | | | | |

n: 0 1 2 3 4 5 6 7

\*/

/\*

2

Write a function called mirror

that takes one parameter a string

and return the reversed string

\*\*hint: "str"[0] => "s"

\*\*hint: "car"[1] => "a"

\*\* Important: You should dint have extra space at the end

mirror("school") => "loohcs"

mirror("car") => "rac"

mirror("maDrasa") => "asarDam"

\*/

/\*

3

Write a function called mirrorCaseAlso

that takes one parameter a string

and return the reversed string order

and the lower case to upper case

and the upper case to lower case

\*\*hint: seacrh in MDN how "a" => "A" and "A" => "a"

\*\* Important: You should dint have extra space at the end

mirrorCaseAlso("sChOol") => "LOoHcS"

mirrorCaseAlso("THOR") => "roht"

mirrorCaseAlso("BaBa") => "AbAb"

\*/

/\*

4

Write a function called repeatChar2

that takes two parameters

and return the times that this char

wit case sensetive

repeat inside this string

\*\*hint: "str"[0] => "s"

\*\*hint: "car"[1] => "a"

\*\*hint: seacrh in MDN how "a" => "A" and "A" => "a"

repeatChar2("schOol","o") => 1

repeatChar2("school","a") => 0

repeatChar2("School","S") => 1

try more case by yourself

\*/

/\*

5

Create a function called HiSayHelloTo

that takes 1 parameter a number that represent

the nubmer of people come to an event,

and say hi to each person sepretadly

+ let him/her say hello to each one that comes before him/her

Ex: welcomeSayHello(3);

=>

hi 1

hi 2 Now Pleas Say Hello To 1

hi 3 Now Pleas Say Hello To 1 and 2

Ex: welcomeSayHello(7); =>

hi 1

hi 2 Now Pleas Say Hello To 1

hi 3 Now Pleas Say Hello To 1 and 2

hi 4 Now Pleas Say Hello To 1, 2 and 3

hi 5 Now Pleas Say Hello To 1, 2, 3 and 4

hi 6 Now Pleas Say Hello To 1, 2, 3, 4 and 5

hi 7 Now Pleas Say Hello To 1, 2, 3, 4, 5 and 6

\*/