

# 1-oy 8-dars. Funksiyalar va ularning turlari HOMEWORK

## HOME WORK

### 1. Array of Multiples

Create a function that takes two numbers as arguments (`num`, `length`) and returns an array of multiples of `num` until the array length reaches `length`.

Function yarating va u 2 ta number typedagi paramert olsin (`num`, `length`) va bizga `length` qiymati qancha bo'lsa shunga array qaytarsin birinchi bergan arrayimizga o'zini qayta qayta qo'shib

#### Examples

```
arrayOfMultiples(7, 5) → [7, 14, 21, 28, 35]

arrayOfMultiples(12, 10) → [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

arrayOfMultiples(17, 6) → [17, 34, 51, 68, 85, 102]
```

### 2. Reverse Words in a String

Given an input string, reverse the string word by word, the first word will be the last, and so on.

Function yarating u **string** typedagi parametrt qabul qilsin va uni teskarisiga qaytarib bersin.

Agarda son berilsa sonlarni ham tekarsiga qaytaradigan bo'lsa zo'r bo'lardi

#### Examples

```
reverseWords(" the sky is blue") → "blue is sky the"

reverseWords("hello world! ") → "world! hello"

reverseWords("a good example") → "example good a"
```

### 3. Burglary Series (01): Calculate Total Losses

You just returned home to find your mansion has been robbed! Given an **object** of the stolen items, return the total amount of the burglary (number). If nothing was robbed, return the string `"Lucky you!"`.

Function yarating u **object** typedagi ma'lumot qabu qisin va objecting parametrlarining qiymatlarni yig'indisini qaytarsin!. Agarda object bo'sh bo'lsa **Lucky you** ni qaytarsin

#### Examples

```
const stolenItems = {
  tv: 30,
  skate: 20,
  stereo: 50,
} → 100

const stolenItems = {
  painting: 20000,
} → 20000

const stolenItems = {} → "Lucky you!"
```

### 4. Remove the Letters ABC

Create a function that will remove the letters "a", "b" and "c" from the given string and return the modified version. If the given string does not contain "a", "b", or "c", return `null`.

Function yarating u **String** paramert qabul qilsin va stringnig ichidagi **a**, **b** va **c** larni o'chirib o'rninga **" "** bo'sh joy qo'ysin. Agarda a,b,c lar yo'q bo'lsa **null** ni qaytarsin

#### Examples

```
removeABC("This might be a bit hard") → "This might e it hrd"

removeABC("hello world!") → null

removeABC("") → null
```

### 5. Capitalize the Names

Create a function that takes an array of names and returns an array where only the first letter of each name is capitalized.

Function yarating u **Array** typedagi qiymatni qabul qilsin va arraydagi hamma elementlarning birinchi harifini katta harfga o'zgartirsin!.

#### Examples

```
capMe(["mavis", "senaida", "letty"]) → ["Mavis", "Senaida", "Letty"]

capMe(["samuel", "MABELLE", "letitia", "meridith"]) → ["Samuel", "Mabelle", "Letitia", "Meridith"]
```

```
capMe(["Slyvia", "Kristal", "Sharilyn", "Calista"]) → ["Slyvia", "Kristal", "Sharilyn", "Calista"]
```

## 6. Find Unique Positive Numbers from Array

Write a function that takes an array and returns a new array with unique positive (more than 0) numbers

*Function yarating u array qabul qilsin u bitta arrayda faqat 1 marotaba qaytarilgan arraylarni qaytarish kerak*

### Examples

```
uniqueArr([-5, 1, -7, -5, -2, 3, 3, -5, -1, -1]) → [1, 3]

uniqueArr([3, -3, -3, 5, 5, -6, -2, -4, -1, 3]) → [3, 5]

uniqueArr([10, 6, -12, 13, 5, 5, 13, 6, 5]) → [10, 6, 13, 5]
```

## 7. Double Factorial

Create a function that takes a number `num` and returns its **double factorial**.

*Function yarating u **number** typedagi paramater qabul qilsin uni faqat **toq numberlar**ning ko'paytmasini recursion bilan qiling*

### Examples

```
doubleFactorial(0) → 1

doubleFactorial(2) → 2

doubleFactorial(9) → 945
// 9*7*5*3*1 = 945

doubleFactorial(14) → 645120
```

## 8. Multiplying Numbers in a String

Given a string of numbers separated by a **comma and space**, return the product of the numbers.

*Function yarating u **String** typedagi numberlar qabul qilsin uni har birini bir biriga ko'paytmasini qiymatini toping!.*

### Examples

```
multiplyNums("2, 3") → 6

multiplyNums("1, 2, 3, 4") → 24

multiplyNums("54, 75, 453, 0") → 0

multiplyNums("10, -2") → -20
```

## 9. Reversible Inclusive List Ranges

Write a function that, given the `start` and `end` values, returns an array containing all the numbers **inclusive** to that range. See examples below.

*Function yarating u **Number** typedagi 2ta parametr qabul qilsin va shu sonlar oralig'idagi sonlar bilan o'sish tartibida arrayga joylab qaytaring!.*

### Examples

```
reversibleInclusiveList(1, 5) → [1, 2, 3, 4, 5]

reversibleInclusiveList(2, 8) → [2, 3, 4, 5, 6, 7, 8]

reversibleInclusiveList(10, 20) → [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

reversibleInclusiveList(24, 17) → [24, 23, 22, 21, 20, 19, 18, 17]
```

## 10. Factorize a Number

Create a function that takes a number as its argument and returns an array of all its factors.

*Function yarating u **number** typedagi parametr qabul qilsin va o'sha numberga karraali bo'lgan hamma sonlarni array ichida qaytarsin.*

### Examples

```
factorize(12) → [1, 2, 3, 4, 6, 12]

factorize(4) → [1, 2, 4]

factorize(15) → [1, 3, 5, 15]

factorize(17) → [1, 17]
```