

13-dars Set, Map HOMEWORK

HOME WORK

- Function yarating u string qiymatni qabul qilsin ya'ni pastdafi `input`ni va natijani `result` kabi qaytarsin
 - Har bir so'zni alohida qaraytaring.
 - Agar so'z uzunligi 3 yoki undan ko'p bo'lsa, so'zni birinchi harfi, keyin so'zning uzunligi minus ikki (chunki birinchi harf va oxirgi harfdan tashqari barcha harflar hisobga olinadi), va nihoyat so'zning oxirgi harfi bilan almashtirish kerak.
 - Agar so'z uzunligi 2 yoki undan kam bo'lsa, so'z o'zgarishsiz qoladi.

Every = E3y bu degani So'zni Birinchi harifi va ohirgisidan bitta oldingi harfgacha uzunligi va ohirgi harf

"Nodejs" ⇒ N4j

"Toshkent" ⇒ T6t

```
const input = "Every developer likes to mix kubernetes and javascript";
```

```
const result = "E3y d7r l3s to mix k8s and j8t";
```

- `students` setidan har bir talabaning o'rtacha ballarini hisoblang va shundan so'ng eng yuqori o'rtacha ballga ega bo'lgan talaba(talabalar)ni qaytaring. Agar bir nechta talabalar eng yuqori o'rtacha ballga ega bo'lsa, ularning barchasi natijada qaytaring

```
const students = new Set([
  { name: "Alice", scores: [90, 85, 92] },
  { name: "Bob", scores: [75, 80, 85] },
  { name: "Charlie", scores: [90, 95, 85] },
  { name: "Jack", scores: [100, 100, 100] }
]);
```

Result

```
[
  { name: 'Jack', average: 100 }
]
```

3. Yaratiladigan funksiya `employees` setini parametr sifatida qabul qiladi. Bu setda har bir xodim haqida ma'lumotlar mavjud bo'lib, ularning ismi, maoshi va departamenti keltirilgan.
- Agar `employees` seti bo'sh bo'lsa, funksiya hatolik xabarini qaytarishi kerak.
 - Sizning vazifangiz `HR` va `IT` departamentlarida ishlaydigan xodimlarning maoshlarini hisoblashdan iborat.
 - Hisoblangan maoshlar yig'indisidan keyin, qaysi departamentda maoshlar o'rtacha ko'proq bo'lsa, o'sha departament haqidagi ma'lumotlarni qaytaring. Natijada faqat eng yuqori o'rtacha maoshga ega departament haqidagi ma'lumotlar bo'lishi kerak.

Input

```
const employees = new Set([
  { name: "John", salary: 50000, department: "IT" },
  { name: "Jane", salary: 60000, department: "HR" },
  { name: "Bob", salary: 55000, department: "IT" },
  { name: "Sophie", salary: 75000, department: "HR" },
  { name: "Mike", salary: 65000, department: "IT" },
  { name: "Emily", salary: 80000, department: "HR" },
  { name: "David", salary: 70000, department: "IT" },
])
```

Result

```
[
  { department: 'HR', average: 71666 }
]
```

4. Eng Ko'p Tarqalgan Tillarni Topish:

Mamlakatlar ma'lumotlaridan foydalanib, eng ko'p tarqalgan tillarni topish uchun maxsus funksiya yarating. Bu funksiya ikkita parametr qabul qiladi: birinchi parametr sifatida mamlakatlar to'plami (Set), ikkinchisi sifatida son qiymati, ya'ni topish kerak bo'lgan

```
let countries = new Set([
  { Russian: 9 },
  { English: 91 },
  { French: 45 },
  { Spanish: 24 },
  { Portuguese: 9 },
  { Dutch: 8 },
  { German: 7 },
  { Arabic: 25 },
  { Chinese: 5 },
  { Swahili: 4 },
  { Serbian: 4 }
])

// Your output should look like this
console.log(mostSpokenLanguages(countries, 3))
```

Result

```
[
  {English:91},
  {French:45},
  {Arabic:25}
]
```

5. Vazifa: Kitoblar Tizimini Boshqarish

Maqsad:

`createBookCollection` obykti yordamida kitoblar to'plamini boshqarish tizimini yarating. Bu tizim quyidagi imkoniyatlarga ega bo'lishi kerak:

- Yangi kitob qo'shish
- Ma'lum bir kitobni ID orqali qidirish
- Barcha kitoblarni ko'rish
- Muallif nomi bo'yicha kitoblarni saralash
- O'qilgan kitoblarni ko'rish
- Ma'lum bir kitobni o'chirish

- Barcha kitoblarni o'chirish

Vazifa Tavsifi:

1. Yangi Kitob Qo'shish (`create`): Yangi kitobni `books` ro'yxatiga qo'shadi. Har bir kitob uchun takrorlanmas `id` yaratiladi.

```
{
  title: 'The Hobbit',
  author: 'J.R.R. Tolkien',
  read: false
}
```

- Kitobni Qidirish (`findOne`): Berilgan `id` orqali kitobni topib, uni qaytaradi.
- Barcha Kitoblarni Ko'rish (`findAll`): Tizimdagi barcha kitoblarni qaytaradi.
- Muallif Bo'yicha Saralash (`filterByAuthor`): Berilgan muallif nomi bo'yicha barcha tegishli kitoblarni qaytaradi.
- O'qilgan Kitoblarni Ko'rish (`filterByreaded`): `read` xususiyati `true` bo'lgan barcha kitoblarni qaytaradi.
- Kitobni O'chirish (`deleteOne`): Berilgan `id` orqali kitobni topib, uni ro'yxatdan o'chiradi.
- Barcha Kitoblarni O'chirish (`deleteMany`): Tizimdagi barcha kitoblarni o'chiradi.

Albatta Map dan foydalanish shart!

- 6.
7. Funksiyani E'lon Qilish: `afterNYears` nomli funksiya yaratishingiz kerak. Bu funksiya ikkita argument qabul qiladi:
- `people`: Bu argument odamlar va ularning hozirgi yoshlarini o'z ichiga olgan obyekt.
 - `n`: Necha yil o'tishi kerakligini ko'rsatuvchi son.
8. Yoshni Hisoblash: Funksiya har bir odamning yoshini `n` yil o'tgach qancha bo'lishini hisoblashi kerak. Agar `n` manfiy son bo'lsa, uning mutlaq qiymati ishlatiladi.
9. Natijani Qaytarish: Funksiya yangilangan yoshlarni o'z ichiga olgan yangi obyektни qaytarishi kerak.

Natija

```
// Misollar
console.log(afterNYears({
  "Joel" : 32,
  "Fred" : 44,
  "Reginald" : 65,
  "Susan" : 33,
  "Julian" : 13
}, 1));
// Natija: { "Joel" : 33, "Fred" : 45, "Reginald" : 66, "Susan" : 34, "Julian" : 14 }
```

```

console.log(afterNYears({
  "Baby" : 2,
  "Child" : 8,
  "Teenager" : 15,
  "Adult" : 25,
  "Elderly" : 71
}, 19));
// Natija: { "Baby" : 21, "Child" : 27, "Teenager" : 34, "Adult" : 44, "Elderly" : 90 }

console.log(afterNYears({
  "Genie" : 1000,
  "Joe" : 40
}, 5));
// Natija: { "Genie" : 1005, "Joe" : 45 }

```

7. O'g'irlangan Buyumlar Ro'yxatidan Biror Narsani Olib Tashlash

Vazifa Tavsifi:

Sizning turmush o'rtog'ingiz sizga ro'yxatdagi buyumlardan biri o'g'irlanmaganini, balki Transilvaniyadagi qasringizda ekanligini aytadi. O'g'irlangan buyumlar obyekti va bir buyum nomi berilganida, ushbu buyumsiz ro'yxatning nusxasini qaytaring.

Misollar:

plaintext

```

removeEntry({ piano: 300, tv: 100, skate: 50 }, "skate") → { piano: 300, tv: 100 }
removeEntry({ mirror: 500, painting: 1 }, "mirror") → {
  painting: 1 }

```

8. Sizning Ismingiz Qanchalik Yaxshi?

Ism qabul qilib, berilgan ismning qanchalik yaxshi ekanligini tekshiradigan funksiya yarating. Harflar uchun ballar lug'ati oldindan yuklangan va Kod tabida mavjud. Ismingizdagi harflarning ballarini qo'shib, jami ballni oling.

```

const scores = new Set([
  ["A", 100],
  ["B", 14],
  ["C", 9],
  ["D", 28],
  ["E", 145],
  ["F", 12],

```

```
["G", 3],  
["H", 10],  
["I", 200],  
["J", 100],  
["K", 114],  
["L", 100],  
["M", 25],  
["N", 450],  
["O", 80],  
["P", 2],  
["Q", 12],  
["R", 400],  
["S", 113],  
["T", 405],  
["U", 11],  
["V", 10],  
["W", 10],  
["X", 3],  
["Y", 210],  
["Z", 23],  
  
])
```

Natijangizni quyidagi qoidalar asosida qaytaring:

$\text{ball} \leq 60$: "NOT TOO GOOD"

$61 \leq \text{ball} \leq 300$: "PRETTY GOOD"

$301 \leq \text{ball} \leq 599$: "VERY GOOD"

$\text{ball} \geq 600$: "THE BEST"

Misol:

`nameScore("MUBASHIR")` → "THE BEST"

`nameScore("YOU")` → "VERY GOOD"

`nameScore("MATT")` → "THE BEST"

`nameScore("PUBG")` → "NOT TOO GOOD"

Bonus

G'olib Kim?

Siz va turmush o'rtog'ingiz o'rtasidagi jang tugadi. Jang qanday o'tganligi haqidagi tasavvuringizga asoslanib, kim g'olib chiqqanini aniqlang.

Uch raunddan iborat obyekt beriladi, har bir raund uchun sizning ochkolariniz ichki obyektlar sifatida keltirilgan. Eng ko'p raundlarda g'olib chiqqan shaxsni hisoblab, g'olibni aniqlang. Raundning g'olibi shu raundda eng ko'p ochko olgan shaxs hisoblanadi. Raundlarda ham, yakuniy natijada ham durang natijalar mumkin.

- Agar siz turmush o'rtog'ingizdan ko'proq raundlarda g'olib chiqqan bo'lsangiz, "MEN!" deb qaytaring.
- Agar turmush o'rtog'ingiz ko'proq raundlarda g'olib chiqqan bo'lsa, "TURMUSH O'RTOG'IM!" deb qaytaring.
- Agar durang bo'lsa, "HECH KIM!" deb qaytaring.

Misol:

javascript

```
let map1 = new Map([
  ["round1", { me: 10, spouse: 5, }],
  ["round2", { me: 5, spouse: 20, }],
  ["round3", { me: 10, spouse: 10, }],
])

let map2 = new Map([
  ["round1", { me: 40, spouse: 5, }],
  ["round2", { me: 9, spouse: 10, }],
  ["round3", { me: 9, spouse: 10, }],
])

determineWinnerOfFight(map1) // → "HECH KIM!"
determineWinnerOfFight(map2) // → "TURMUSH O'RTOG'IM!"
```