

Link: <https://donmezyusuf.github.io/Advanced-Programming-Techniques/HW2/yusufhw2.html>

1. Soru

a-) Verilen öğrenci numarası ile öğrencinin sınav takvimini bulma

Student Database

Homework

Given Student:

Given Course:

Given Exam Room:

Given Room:

Given GPA:

Student id:

Course id:

Result:

3128 items

211561165 Numaralı öğrencinin sınav takvimi:

ENGR 244 dersi saat, tarih ve derslik formatında olmak üzere:

12:00-15:00 15.01.2017 B1039 B1038 B1037 da yapılacaktır.

ENGR 251 dersi saat, tarih ve derslik formatında olmak üzere:

15:00-18:00 12.01.2017 BZ020 BZ021 BZ022 BZ023 da

yapılacaktır.

LIFE 101 dersi saat, tarih ve derslik formatında olmak üzere: 12:00-

15:00 14.01.2017 BZ003 BZ004 BZ005 BZ006 BZ007 BZ008

BZ009 BZ010 BZ011 BZ012 BZ013 da yapılacaktır.

b-) Verilen ders isminden öğrenci listesini bulma

Student Database

Random

Best GPA

Clear

Homework

Given Student:

Give Exam schedule

Given Course:

ARAB 102

Give Student list

Given Exam Room:

Give Course list

Given Room:

Give Total Number of Course

Given GPA:

Give Student List from GPA

2. Question Solution Run

Student id:

Course id:

math 206

ALİ *211552207*

ESLEM *211561003*

ENES *212061635*

YUSMAR *212171715*

RUMEYSA *213010951*

KERİME *214052395*

NAZIM *214101760*

HAVVANUR *216400553*

MUSA *514250060*

c-) Verilen derslik numarasından o derslikteki ders listesi bulma

Student Database

Random

Best GPA

Clear

Homework

Given Student:

Give Exam schedule

Given Course:

Give Student list

Given Exam Room:

Give Course list

Given Room:

Give Total Number of Course

Given GPA:

Give Student List from GPA

2. Question Solution Run

Student id:

Course id:

ADP 105

ARAB 101 02-03

BGM 501

BGM 502

BGM 539

BGM 551

BGM 553

BGM 555

BPP 101

BPP 107

CGP 101

EE 301

EE 311

HUK 405

MAT 103

MGT 453

OTM 101

PHIL 225

POLS 203

RUS 101

d-) Verilen derslik isminden o derslikteki toplam dersleri bulma

Student Database

Random

Best GPA

Clear

Homework

Given Student:

Give Exam schedule

Given Course:

Give Student list

Given Exam Room:

Give Course list

Given Room:

Give Total Number of Course

Given GPA:

Give Student List from GPA

2. Question Solution Run

Student id:

Course id:

BTB02 sınıfındaki sınav sayısı: 20

e-) Verilen GPA'ye sahip öğrencileri listeleme

Student Database

Random

Best GPA

Clear

Homework

Given Student:

Give Exam schedule

Given Course:

Give Student list

Given Exam Room:

Give Course list

Given Room:

BTB02

Give Total Number of Course

Given GPA:

2.20

Give Student List from GPA

2. Question Solution Run

Student id:

Course id:

math 206

GPA'i 2.20 olan öğrencileri

Sample code

```
function giveExamSchedule() {
  let scheduleArray = []

  let studentId = document.getElementById("studentId").value
```

BERRA
MERYEM
YELİZ
HANNE
ALİ
BERKAY
TALHA
ABDULKERİM
RAMAZAN
BARIŞ

2.Soru -) stdMap'te olan öğrencilerin keylerini keys isimli bir array'e aktarıldı. Daha sonra random 9 basamaklı bir sayı üretildi. Map.has ve keys.includes için ayrı ayrı bir for döngüsü yazıldı. Bu for döngülerinde oluşturduğumuz rastgele sayı birer arttırılarak öğrencilerin key'leri ile eşleşme sağlanıp sağlanmadığı kontrol edildi eğer eşleşme olursa counter arttırıldı. Bu kontrol başlamadan önce for döngüsünde mevcut zamanı bir flag ile tutuldu, sonra for bitince geçen zamanı öğrenmek için başlangıç zamanından bitiş zamanını çıkarıldı.

Student Database

Homework

Given Student:

Given Course:

Given Exam Room:

Given Room: BTB02

Given GPA: 2.20

Student id:

Course id:

iki yöntem arasındaki fark msec:29 isabetli değer: 1
iki yöntem arasındaki fark msec:31 isabetli değer: 1
iki yöntem arasındaki fark msec:49 isabetli değer: 0
iki yöntem arasındaki fark msec:31 isabetli değer: 0
iki yöntem arasındaki fark msec:28 isabetli değer: 0

Kaynak Kod

Sample code

```
function giveExamSchedule(){
    let studentId= document.getElementById("studentInput").value;
    if(stdMap.has(studentId))
    {
        let studentLessons = stdMap.get(studentId).list
        report(studentId + " Numaralı öğrencinin sınav takvimi:")
        report("")
        for (let i = 0; i < studentLessons.length; i++) {
            let lesson = studentLessons[i];
            report(lesson + " dersi saat, tarih ve derslik formatında olma
            crsMap.get(lesson).data + " da yapılacaktır.\n")
        }
    }
}

function giveStudentList(){
    let namesArray = [];
    let courseId= document.getElementById("courseInput").value;
    if(crsMap.has(courseId))
    {
        for (let val of stdMap.values()) {
            if(val.list.includes(courseId)){
                namesArray.push(val.name + " " + val.id);
                report( val.name,val.id);
            }
        }
    }
}

function giveCourseList(){
    let courseArray = []

    let examId= document.getElementById("examInput").value;
    for (let val of crsMap.values()) {
        let splitted= val.data.split(" ")
        if(splitted.includes(examId) && examId !== ""){
            courseArray.push(val.id);
            report(val.id);
        }
    }
}

function giveTotalNumberOfCourse(){
    let courseArray = []

    let roomId=document.getElementById("roomInput").value;
    let count = 0;
    for (let val of crsMap.values()) {
        let splitted= val.data.split(" ")
        if(splitted.includes(roomId) && roomId !== ""){
            courseArray.push(val.id);
            count++
        }
    }
    report(roomId+" sınıfındaki sınav sayısı: " , count,courseArray)
}
```

```

function giveStudentListFromGPA(){
    let studentArray = []
    let id = document.getElementById("gpaInput").value;
    id = Number(id).toFixed(2)
    for(let x of stdMap.values()){
        if(id === x.gpa && id !== ""){
            studentArray.push(x.name)
        }
    }
    report("GPA'i "+id+" olan", "öğrenciler",studentArray)
}
function benchmark(){
    let randomID= Math.floor(Math.random() * 1000000000);
    let keys = []
    let countForKey = 0, countForMap = 0;
    let endFlagForKey= 0;
    let endFlagForMap = 0;
    for(let std of stdMap){
        keys.push(std[0])
    }
    let startFlagForKey = Date.now();
    for (let i = 0; i < 10000; i++) {
        if(keys.includes(randomID + i)){
            countForKey++;
        }
    }
    endFlagForKey = Date.now() - startFlagForKey;

    let startFlagForMap = Date.now();
    for(let t= 0; t <10000; t++){
        if(stdMap.has(randomID + t)){
            countForMap++;
        }
    }
    endFlagForMap = Date.now() - startFlagForMap;
    report("iki yöntem arasındaki fark msec:" + endFlagForKey +
    " isabetli değer: " + endFlagForMap )
}

```
