

Mesut Yılmaz - 1721221220

<https://mesutyilmaz01.github.io/Apt/%C3%96devler/Homework2/homework2.html>

Random Best GPA Clear

Exam schedule for a given student:

716363208 Get !

Student list taking a given course:

IE 316 Get !

Course list for a given exam room:

BZ015 Get !

Total number of courses in a given room:

BZ015 Get !

Course list for a given date:

13.01.2017 Get !

Compare Two Method ! (Second Side)

Result:

262 items

3128 items

716363208 Numaralı Öğrencinin Sınav Takvimi:

Ders : IE 316 Tarih: 12.01.2017 Saat: 12:00-15:00

Ders : IE 462 Tarih: 6.01.2017 Saat: 14:00-17:00

Ders : ISE 552 Tarih: 6.01.2017 Saat: 14:00-17:00

Ders : ITM 301 Tarih: 9.01.2017 Saat: 09:00-12:00

Ders : MGT 203 Tarih: 12.01.2017 Saat: 15:00-18:00

Ders : MGT 302 Tarih: 2.01.2017 Saat: 12:00-15:00

Ders : MGT 305 Tarih: 8.01.2017 Saat: 09:00-12:00

Ders : OTM 101 Tarih: 11.01.2017 Saat: 12:00-15:00

Bu Dersi Alan Öğrenciler :

ID : 210120699 Adı : HAMİT Gpa : 1.52  
ID : 211580024 Adı : BERKAY Gpa : 2.20  
ID : 211591387 Adı : CİHAD Gpa : 1.08  
ID : 211760265 Adı : ALİ Gpa : 3.59  
ID : 212030264 Adı : EMRE Gpa : 3.64  
ID : 212061274 Adı : SEVGİ Gpa : 2.23  
ID : 212110046 Adı : ENES Gpa : 2.46  
ID : 212331725 Adı : FATHİ Gpa : 1.94  
ID : 212470756 Adı : YASEEN Gpa : 3.25  
ID : 212980975 Adı : GAZİ Gpa : 1.33  
ID : 212990683 Adı : AYBÜKE Gpa : 1.97  
ID : 212991303 Adı : TAHA Gpa : 3.27  
ID : 213052571 Adı : İBRAHİM Gpa : 2.31  
ID : 213120580 Adı : TANSU Gpa : 3.59  
ID : 213162775 Adı : OBADA Gpa : 2.34  
ID : 213172755 Adı : SAEED Gpa : 2.86  
ID : 213911529 Adı : YASİN Gpa : 3.19  
ID : 213950676 Adı : TALHA Gpa : 3.94  
ID : 514583215 Adı : FATİH Gpa : 3.37  
ID : 716363208 Adı : ENES Gpa : 1.47

Bu Sınıfta Sınavı Olacak Dersler :

Ders Kodu : ADP 103 Tarihi : 10.01.2017 Saati :09:00-12:00  
Ders Kodu : ADP 107 Tarihi : 4.01.2017 Saati :12:00-15:00  
Ders Kodu : CHI 101 Tarihi : 11.01.2017 Saati :12:00-15:00  
Ders Kodu : CS 351 Tarihi : 2.01.2017 Saati :12:00-15:00  
Ders Kodu : CS 371 Tarihi : 5.01.2017 Saati :15:00-18:00  
Ders Kodu : EE 312 Tarihi : 12.01.2017 Saati :12:00-15:00  
Ders Kodu : HIST 307 Tarihi : 14.01.2017 Saati :15:00-18:00  
Ders Kodu : HIST 593 Tarihi : 14.01.2017 Saati :15:00-18:00  
Ders Kodu : HUK 101 Tarihi : 6.01.2017 Saati :14:00-15:00  
Ders Kodu : IE 451 Tarihi : 3.01.2017 Saati :15:00-18:00  
Ders Kodu : ISS 209 Tarihi : 3.01.2017 Saati :09:00-12:00  
Ders Kodu : LAW 213 Tarihi : 5.01.2017 Saati :09:00-12:00  
Ders Kodu : LAW 321 Tarihi : 8.01.2017 Saati :09:00-12:00  
Ders Kodu : LING 431 Tarihi : 14.01.2017 Saati :15:00-18:00  
Ders Kodu : MATH 206 Tarihi : 11.01.2017 Saati :09:00-12:00  
Ders Kodu : MGT 203 Tarihi : 12.01.2017 Saati :15:00-18:00  
Ders Kodu : MGT 303 Tarihi : 8.01.2017 Saati :15:00-18:00

Bu sınıfta verilen toplam ders sayısı : 24

Girilen Tarihte Sınavı Olacak Dersler :

Ders Kodu : ADP 105 Tarihi : 13.01.2017 Saati :14:00-17:00

Ders Kodu : BGM 543 Tarihi : 13.01.2017 Saati :18:00-21:00

Ders Kodu : CTV 253 Tarihi : 13.01.2017 Saati :14:00-17:00

Ders Kodu : ECON 201 Tarihi : 13.01.2017 Saati :09:00-12:00

Ders Kodu : IE 324 Tarihi : 13.01.2017 Saati :14:00-17:00

Ders Kodu : ISS 475 Tarihi : 13.01.2017 Saati :14:00-17:00

Ders Kodu : LAT 101 Tarihi : 13.01.2017 Saati :09:00-12:00

Ders Kodu : MGT 255 Tarihi : 13.01.2017 Saati :14:00-17:00

Ders Kodu : POLS 219 Tarihi : 13.01.2017 Saati :09:00-12:00

Ders Kodu : PSY 231 Tarihi : 13.01.2017 Saati :14:00-17:00

---

Has metodu ile çıkan sonuç : 56

Include metodu ile çıkan sonuç : 56

Has metodu ile algortima zamanı : 6

Include metodu ile alogirtma zamanı : 130

---

Has metodu ile çıkan sonuç : 56

Include metodu ile çıkan sonuç : 56

Has metodu ile algortima zamanı : 4

Include metodu ile alogirtma zamanı : 131

---

## Sample code

```
class Course { //implement this
  constructor(id,time,date,...list) {
    if(!id) throw "Undefined Course id"
    this.id = id;
    this.time = time;
    this.date = date;
    this.list = list;
  }
  toString() {
    return this.id+' '+'+'+this.time+' '+this.date;
  }
  static fromString(s) {
    return new Course(...s.split('\t'));
  }
}

function readCourses() {
  fetch(LINK+"Courses.txt").then(r => r.text())
  .then(t => makeData(t, Course, crsMap))
}

function findBest() { //the best gpa in stdMap
  let bestgGpa = 0
  for (let b of stdMap.values()){
    if(b.gpa > bestgGpa){
      bestgGpa = b.gpa
    }
  }
  report(bestgGpa);
}

function courseStudentList(){
  let value = document.getElementById("texttwo").value.toUpperCase();
  let counter = 0;
  for(let student of stdMap.values()){
    if(student.list.includes(value)){
      counter++;
      if(counter==1) report("Bu Dersi Alan Öğrenciler :");
      report("ID : "+student.id+" Adı : "+student.name+" Gpa : "+student.gpa);
    }
  }
}

function examRoomCourseList(){
  let value = document.getElementById("textthree").value.toUpperCase();
  let counter = 0;
  for(let course of crsMap.values()){
    if(course.list.includes(value)){
      counter++;
      if(counter==1) report("Bu Sınıfta Sınavı Olacak Dersler :");
      report("Ders Kodu : "+course.id+" Tarihi : "+course.date+" Saati :"+course.time);
    }
  }
}

function givenRoomTotalNumberCourse(){
  let counter = 0;
  let value = document.getElementById("textfour").value.toUpperCase();
  for(let courses of crsMap.values()){
    if(courses.list.includes(value)){
      counter++;
    }
  }
  if(counter == 0){
    report("Bu sınıfta ders yoktur");
  }else{
    report("Bu sınıfta verilen toplam ders sayısı : "+counter);
  }
}

function myQuery(){
  let value = document.getElementById("textfive").value.toUpperCase();
  let counter = 0;
  for(let courses of crsMap.values()){
    if(courses.date == value){
      counter++;
      if(counter==1) report("Girilen Tarihte Sınavı Olacak Dersler :");
      report("Ders Kodu : "+courses.id+" Tarihi : "+courses.date+" Saati :"+courses.time);
    }
  }
}
```



```

function studentExamSchedule(){
    let courses = [];
    let value = document.getElementById("textone").value;
    let std = stdMap.get(value);
    if (!std){
        return report(value+" Numaralı Öğrenci Sistemde Kayıtlı Değil");
    }else{
        for(let i=0; i<std.list.length; i++){
            let temp = crsMap.get(std.list[i].toString());
            if(temp){
                courses.push(temp);
            }else{
                return report(value+" Numaralı Öğrencinin Sistemde Kayıtlı Dersi Bulunamamıştır");
            }
        }
        report(value+" Numaralı Öğrencinin Sınav Takvimi:");
        for(let course of courses){
            report("Ders : "+course.id+" Tarih: "+course.date+" Saat: "+course.time);
        }
    }
}

```

```

function secondSide(){
    let stuID = 116690070;
    let has = 0;
    let include = 0;
    let keys = [];
    for (let number of stdMap) {
        keys.push(number[0])
    }
    let hasTime = Date.now();
    for (let index=stuID; index<stuID+10000; index++) {
        if(stdMap.has(index.toString())){
            has+=1;
        }
    }
    hasTime = Date.now() - hasTime;
    let includeTime = Date.now()
    for (let index=stuID; index<stuID+10000; index++) {
        if(keys.includes(index.toString())){
            include+=1;
        }
    }
    includeTime = Date.now() - includeTime;
    report("<hr>");
    report("Has metodu ile çıkan sonuç : "+has)
    report("Include metodu ile çıkan sonuç : "+include)
    report("Has metodu ile algortima zamanı : "+hasTime)
    report("Include metodu ile alogirtma zamanı : "+includeTime)
    report("</hr>");
}

```

## Açıklamalar

CourseStudentList Fonksiyonu :

İlgili inputdan gelen veri öncelikle arama sonuçları ile eşleşmesi için büyük harfe çevrilir. For döngüsü ile stdMap value kısmı dönülür ve ilgili bilgi burada mevcut ise ekrana bilgiler yazdırılır. Counter kullanımı ise başlık oluşması içindir.

ExamRoomCourseList Fonksiyonu :

İlgili inputdan gelen veriye göre courseMap value kısmı for döngüsü ile dönülür ve eğer ilgili bilgi mevcut ise ekrana yazdırılır. Counter kullanımı başlık içindir.

GivenRoomTotalNumberCourse Fonksiyonu :

İlgili inputdan veri alınır ve map içindeki bilgi ile uyuşması amacı ile büyük harflere çevrilir. Sonrasında for döngüsü ile crsMap value kısmı dönülür ve forun üstünde tanımlanan counter sayısı bilgi uyuşması olursa arttırılır. For döngüsünden sonra counter 0 ise ders yok, değilse counterin kendisi yazdırılır.

MyQuery Fonksiyonu:

İlgili inputdan veri alındıktan sonra for döngüsü ile crsMapin value kısmı dönülür ve alınan veri ile uyuşma sağlanırsa ekrana yazdırılır.

StudentExamSchedule Fonksiyonu :

Öncelikle inputdan gelen veri ile uyuşan öğrenci olup olmadığına bakılır. Eğer yoksa öğrenci mevcut değil yazılır. Varsa bu öğrencinin ders listesi bilgileri for döngüsü ile dönülür ve daha önceden oluşturulmuş boş bir arraye atılır. Ardından bu array dönülerek bilgiler ekrana basılır.

SecondSide Fonksiyonu :

Öncelikle keys adlı boş bir array oluşturulur. Bu arraye stdMap fonksiyonun key kısmı dönülerek her öğrencinin numarası keys adlı arraye aktarılır. Ardından seçilen bir öğrenci numarasından 10000 sayı ileri gidecek şekilde bir for döngüsü oluşturulur ve içinde stdMapin has metodu ile aratma yapılır. Bu for döngüsünün üstüne ve altına zaman alınacak şekilde değişken tanımlanır. Bu değişkenlerin farkı zaman bilgisini vermektedir. Aynı işlem keys arrayi için de yapılır fakat keys arrayi için include methodu kullanılır.

Açıkça görülmektedir ki map fonksiyonu normal bir arraye göre çok daha hızlı çalışmaktadır.