

FINAL EXAM HOMEWORK – FINAL SINAV ÖDEVİ

Bir üniversitede bir bölümde sunulan derslerin verilerini ve bu derslere kayıtlı öğrencilerin kayıtlarını depolamak için aşağıda ayrıntıları verilen makro tanımlarını ve yapı tanımlarını kullanarak bir C programı yazın.

Write a C program using the macro definitions and structure definitions as detailed below to store data for courses offered at a department at a university and the record of students registered to these courses.

```
#define CLASSSIZE 35           // SINIFKAPASITE
#define STUDENTNUMBERLENGTH 10 // OGRENCINUMARAUZUNLUGU
#define NAMELENGTH 25         // ADIUZUNLUGU
#define COURSENAMELENGTH 50   // DERSADIUZUNLUGU
#define COURSECODELENGTH 10   // DERSKODUZUNLUGU
#define COURSESREGISTERED 5    // KAYITLIDERSLER
#define MIDTERMWEIGHT 30      // VIZEAGIRLIK
#define FINALWEIGHT 30        // FINALAGIRLIK
#define QUIZWEIGHT 20         // QUIZAGIRLIK
#define HOMEWORKWEIGHT 20     // ODEVAGIRLIK

typedef struct fullName{      //tam adi
    char *name;               //adi
    char *surname;           //soyadi
}name_t;

typedef struct examGrades{    //sinav not
    double midterm;          //vize
    double final;            //final
    double quiz;             //quiz
    double homework;         //odev
}grades_t;

typedef struct courseRegistered{ //kayitliDersler
    char courseName[COURSENAMELENGTH]; //ders adi
    char courseCode[COURSECODELENGTH]; //ders kodu
    grades_t grades;                   //notler
    char letterGrade[2];               //herf notu
    double numericGrade;               //numerik notu
}courseReg_t;

typedef struct studentInfo{    //ogrenci bilgi
    name_t studentName;       //ogrenci adi
    char *stNo;               //ogrenci numara
    courseReg_t *courses[COURSESREGISTERED]; //dersler
    double gpa;               //ortalama
}student_t;

typedef struct courseInfo{    //ders bilgi
    char courseName[COURSENAMELENGTH]; //ders adi
    char courseCode[COURSECODELENGTH]; //ders kodu
    student_t students[CLASSSIZE];     //ogrenciler
    double courseAverage;               //ders ortalama
}courses_t;
```

Programınızda `ogrenciEkle()`, `dersEkle()`, `OgrencininDersNotuHesapla()`, `OgrenciOrtalamaHesapla()`, `OgrenciBilgiGoster()`, `DersBilgiGoster()`, `SinifGoster()` ve `DersOrtalamaHesapla()` işlevleri gerçekleştirin.

Lütfen herhangi bir global değişken kullanmanıza izin verilmediğini unutmayın. Kullanılan tüm değişkenler yerel değişkenler olmalıdır. Lütfen değişkenleri bir diyagram kullanarak bellekte göründükleri gibi detaylandırın. Lütfen akış şemalarınızı ve teknik açıklamalarınızı iyi belgelenmiş kaynak kod dosyalarıyla birlikte programlara sağlayın.

In your program perform functions to `addStudent()`, `addCourse()`, `calculateStudentCourseGrade()`, `calculateStudentGPA()`, `displayStudentInfo()`, `displayCourseInfo()`, `printClass()` and `calculateCourseAverage()`.

Please remember that you are not allowed to use any global variables. All variables used must be local variables.

Please detail variables as they appear in memory using a diagram. Please provide your flowcharts and technical explanations to the programs along with well documented source code files.

`main()` işlevin içinde, ihtiyacınız olan diğer değişkenlerle birlikte `course_t` `courseClass` değişkenini kullanın; hem sınıf verilerini hem de öğrenci verilerini doldurmak ve bunları görüntülemek için.

Bir öğrencinin herhangi bir sınavdan 0'ın altında bir not veya 100'ün üzerinde bir not alamayacağını ve programınızın kullanıcı tarafından verilen girdinin aralık içinde olduğundan emin olması gerektiğini lütfen unutmayın.

Ayrıca, harf notlarının "AA" 4,00 olduğunu ve 100 - 90 notları için verildiğini, "BA" 3,50 olduğunu ve 85 - 89 notları için verildiğini, "BB" 3,00 olduğunu ve 84 - 80 notları için verildiğini lütfen unutmayın, "CB" 2.50'dir ve 79-75 notu için verilir, "CC" 2.00'dür ve 74 - 70 notu için verilir, "DC" 1.50'dir ve 69 - 60 notu için verilir, "DD" 1.00'dir ve bir 59 - 50 ve "FF" 0,00 olup 49 - 0 notu için verilir. Not ortalaması 0,00 ile 4,00 arasında olmalı ve her zaman doğru biçimde gösterilmelidir.

Inside the `main()` function, along with other variables you need, use the variable `courses_t` `courseClass`; to both populate the class data and student data, as well as to display them.

Please note that a student cannot get a grade below 0 or a grade above 100 for any exam and your program must make sure that input given by user is within the range.

Please also note that letter grades are "AA" is 4.00 and awarded for a grade 100 – 90, "BA" is 3.5 and awarded for a grade 85 – 89, "BB" is 3.00 and awarded for a grade 84 – 80, "CB" is 2.50 and awarded for a grade 79 – 75, "CC" is 2.00 and awarded for a grade 74 – 70, "DC" is 1.50 and awarded for a grade 69 – 60, "DD" is 1.00 and awarded for a grade 59 – 50 and "FF" is 0.00 and awarded for a grade 49 – 0. The GPA must be between 0.00 to 4.00 and should always be displayed in the correct format.

Bilginizi ne kadar iyi sergilediğinize göre derecelendirileceğinizi lütfen unutmayın. Verilen işlevselliğin tamamen uygulanması ve ayrıntılı olması koşuluyla, programınıza başka işlevler eklemekten çekinmeyin. Bir öğrenci

programdaki her şeyi yapamazsa, yine de kısmi bir çalışma yapabilir ve çalışmalarını hem şematik hem de sözlü olarak (bir belge şeklinde) açıklayıp detaylandırmaları koşuluyla notlandırılacaktır. Kısmi çalışmalar kısmi olarak notlandırılacaktır.

Please remember you will be graded based on how well you showcase your knowledge. Feel free to add any further functionality to your program provided the given functionality is completely implemented and detailed.

If a student cannot do everything in the program they can still do partial work and that will be graded provided they can explain and detail their work both schematically and verbally (in form of a document). Partial work will be partially graded.

GOOD LUCK