

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Öğrencinin Adı Soyadı:	Öğrenci No:	İmza:
Dersin Adı: BLM2031 Yapısal Programlama	Tarih/Saat: 24/11/2020 - 09.00	Sınav süresi: 90
Sınav Türü:	Vize 1 +	Vize 2
	Mazeret	Final
		Bütünleme
Unvan Ad-Soyad:	Grup-1 Dr. Öğr. Üyesi Yunus Emre SELÇUK Grup-2 Öğr. Gör. Dr. Ahmet ELBİR	

SORULAR

Not-1: Soruların tümünde, global değişken, döngü içinde break continue, go to kullanımı yasaktır.

Not-2: “_____” gördüğünüz yeri ya boş bırakmanız ya da uygun şekilde doldurmanız gerekecektir

Not-3: Cevaplarınız Soru1.c ve Soru2.c dosyalarını içeren ÖğrenciNo.rar dosyasında online sisteme yüklenmelidir.

- 1) (40p) Kullanıcıdan alınan N sayısına göre N farklı karmaşık sayının reel ve sanal kısmı, Nx2 boyutlu bir **dinamik int** matriste saklanacaktır. Main fonksiyonda bu matrise bilgi girişi yapıldıktan sonra, aşağıdaki işlemleri yapan fonksiyonları yazınız.
- Matrisi parametre olarak alıp reel ve sanal kısımların kareleri toplamı en büyük karmaşık sayının kaçınıcı satırda olduğunu bulan ve geri döndüren fonksiyonu yazınız.
 - Fonksiyon prototipi:** _____ enbuyuk_bul(matris parametre olarak alınacaktır);
 - Matrisi parametre olarak aldıktan sonra N sayıdaki karmaşık sayının her birisini a + bi formatında alt alta yazdıran fonksiyonu yazınız.
 - Fonksiyon prototipi:** _____ liste_yazdir(matris parametre olarak alınacaktır);
 - Örnek Çıktı:**
2 + 3i
5 - 2i
...
7 + 4i
- 2) (60p) Öğrenci bilgi sistemi için id, Ad, yaş ve ortalama bilgilerini saklayan bir struct üzerinde aşağıdaki işlemleri yapan programı yazınız
- id, Ad, yaş ve ortalama bilgileri için gereken struct'ı tanımlayınız
 - Main fonksiyonu içerisinde kullanıcıdan alınan N sayısına göre N elemanlı bir dinamik öğrenci dizisi oluşturunuz.
 - Oluşturulan öğrenci dizisini ortalama göre sıralayan bir fonksiyon yazınız. Sıralama sonrası main içerisinde tanımlanan dizinin orijinal hali değiştirilmelidir. İstedığınız sıralama algoritmasını kullanabilirsiniz.
 - Fonksiyon prototipi:** _____ nota_gore_sirala(struct öğrenci, eleman sayısı);
 - Oluşturulan öğrenci dizisini Ad'a göre sıralayan bir fonksiyon yazınız. Sıralama sonrası main içerisinde tanımlanan dizinin orijinal hali değiştirilmelidir. İstedığınız sıralama algoritmasını kullanabilirsiniz.
 - Fonksiyon prototipi:** _____ ada_gore_sirala(struct öğrenci, eleman sayısı);
 - Oluşturulan öğrenci dizisini ekrana id - ad- yas- ortalama formatında yazdıracak bir fonksiyon yazınız.
 - Fonksiyon prototipi:** _____ liste_yaz(struct öğrenci, eleman sayısı);
 - Main içerisinde oluşturulan öğrenci dizisi için c, d ve e maddelerindeki işlemleri aşağıdaki sıraya göre çağırarak fonksiyon çağrılarını yazınız. Fonksiyon çağrılarından sonra dinamik olarak tahsis edilen bellek bölgesini serbest bırakan kod/kodları yazınız.
 - Listenin ilk halini ekrana yazdıran fonksiyon çağırısı
 - Listeyi ortalama göre sıralayan fonksiyonun çağırısı
 - Listeyi ad'a göre sıralayan fonksiyonun çağırısı
 - Listenin en son halini ekrana yazdıran fonksiyonun çağırısı