**T.C.**

**KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**2022-2023 EĞİTİM ÖĞRETİM YILI GÜZ DÖNEMİ**

**YÜKSEK DÜZEY PROGRAMLAMA DERSİ (TEKNİK SEÇMELİ DERS V) FİNAL BÜTÜNLEME SINAVI**

**18.01.2023, Çarşamba, Saat 09:00**

**Not:** Sadece 3 soru çözülecektir.

Toplam süre 45 dakikadır.

Sorular öğrencide kalabilir.

**Soru 1:** İkinci dereceden bir polinomun köklerini hesaplayan Python programı isteniyor. “KökHesap” altprogramı konsol ekranın en üstüne “2.Dereceden Denklem Köklerinin Hesabı” yazdırıyor. Sonra sırayla polinomun A, B, C katsayılarını her biri için “GirKatsayı” altprogramını kullanarak klavyeden istiyor. Girilen değer geçerli bir reel sayı değilse “X Katsayısı Reel Sayı Olmalı” uyarısı veriyor. X burada A, B ve C katsayılardan biridir. Reel sayı girilene kadar tekrar isteniyor. Katsayılar girildikten sonra polinomun köklerini hesaplıyor. Konsol ekrana kökler reel ise “Kökler Reel”, katlı ise “Kökler Katlı” veya sanal ise “Kökler Sanal” yazdırıyor. Sonra köklerini anlaşılır bir biçimde konsol ekrana yazdırıyor. Gerekli kodları tamamlayınız.

**KökHesap()**

**Soru 2:** Aşağıdaki işi yapan ve dört altprogramdan oluşan bir Python programı isteniyor. “YapDizi” altprogramı eleman sayısını girdi olarak aldığı bir dizi oluşturup elemanlarına 0 ile 999 arasında rasgele tamsayılar atıyor. “YazDizi” altprogramı konsol ekrana “Dizi Elemanları” yazdırdıktan sonra anlaşılır bir biçimde sırayla sıra numaralarıyla beraber dizi elemanlarını yazdırıyor. “OkuTamsayı” altprogramı klavyeden bir girdi istiyor. Girdi tamsayı değilse uyarı veriyor. Tamsayı girilene kadar uyarı tekrarlanıyor. “LinearSearch” altprogramı ise bu girilen tamsayıyı girdi olarak alıp “Doğrusal Arama Algoritması” mantığına göre dizi içinde arıyor. Bulursa indisini, bulamazsa -1 döndürüyor. Ayrıca program konsol ekrana “Doğrusal Arama Algoritması” yazdırdıktan sonra klavyeden girilen tamsayının dizinin kaçıncı elemanı olduğunu söylüyor. Dizi için liste değişkeni kullanılacak. Gerekli kodları tamamlayınız.

**import random**

**a = YapDizi(15)**

**YazDizi(a)**

**x = OkuTamsayı()**

**i = LinearSearch(a, x)**

**print("Doğrusal Arama Algoritması")**

**if i >= 0: print(f"{x:5} : Dizinin {i+1}.elemanı")**

**else: print(f"{x:5} : Dizi elemanı değil")**

**Soru 3:** Aşağıdaki işi yapan ve üç altprogramdan oluşan bir Python programı isteniyor. “YapDizi” altprogramı eleman sayısını girdi olarak aldığı bir dizi oluşturup elemanlarına 0 ile 999 arasında rasgele tamsayılar atıyor. “YazDiziler” altprogramı dizi elemanları normal sırasında, artan sırada ve azalan sırada olmak üzere başlıkları ile beraber üç sütün şeklinde konsol ekrana yazdırıyor. “BubbleSort” altprogramı “Kabarcık Sıralama Algoritması” mantığına göre diziyi küçükten büyüğe sıralıyor. Ayrıca program dizinin eleman sayısını ve altına “Kabarcık Sıralama Algoritması” yazısını yazdırıyor. Dizi için liste değişkeni kullanılacak. Gerekli kodları tamamlayınız.

**import random**

**a = YapDizi(15)**

**b = a.copy()**

**BubbleSort(b)**

**c = b.copy()**

**c.reverse()**

**YazDiziler(a, b, c)**

**print(f"Eleman Sayısı : {len(a)}")**

**print("Kabarcık Sıralama Algoritması")**

**Soru 4:** A ve B matrislerini oluşturup çarpan ve sonucu C matrisine atayan bir Python programı isteniyor. Ayrıca bu program matris boyutlarını hazır olarak alıyor ve matrisleri anlaşılır bir biçimde ekrana yazdırıyor. Program üç altprogram içeriyor. “YapMatris” altprogramı boyutlarını girdi olarak aldığı bir matris oluşturup her bir elemanına -50 ile +50 arasında rasgele reel sayılar atıyor. “YazMatrisEkran” altprogramı ismini girdi olarak aldığı matrisin boyutlarını ve matris formunda içeriğini konsol ekrana yazdırıyor. Yazdırıp yazdıramadığını konsol ekranda belirtiyor. “CarpMatris” altprogramı iki matris çarpıp yeni bir matris oluşturuyor. Matrisler için liste değişkeni kullanılacak. Gerekli kodları tamamlayınız.

**import random**

**m, n, p = 5, 4, 3**

**a = YapMatris(m, p)**

**b = YapMatris(p, n)**

**c = ÇarpMatris(a, b)**

**print(f"{m}\*{p}'lik A Matrisi")**

**YazMatrisEkran(a)**

**print(f"{p}\*{n}'lik B Matrisi")**

**YazMatrisEkran(b)**

**print(f"{m}\*{n}'lik C = A\*B Matrisi")**

**YazMatrisEkran(c)**

**YÜKSEK DÜZEY PROGRAMLAMA DERSİ (TEKNİK SEÇMELİ DERS V) FİNAL SINAVI**

**Adı Soyadı :**

**Numarası :**

**Çözüm 1**

**def GirKatsayı(s):**

**while True: 3**

**try: 2**

**return float(input(s + " : ")) 3**

**except ValueError: 2**

**print(s, "Reel Sayı Olmalı") 2**

**def KökHesap():**

**print("2.Dereceden Denklem Köklerinin Hesabı") 2**

**a = GirKatsayı("A Katsayısı") 2**

**b = GirKatsayı("B Katsayısı") 2**

**c = GirKatsayı("C Katsayısı") 2**

**delta = b \* b - 4 \* a \* c 3**

**if delta < 0: print("\nKökler Sanal") 3**

**elif delta > 0: print("\nKökler Reel") 3**

**else: print("\nKökler Katlı") 3**

**x1 = (-b - delta \*\* 0.5) / (2 \* a) 3**

**x2 = (-b + delta \*\* 0.5) / (2 \* a) 3**

**print(f"x1 = {x1}\nx2 = {x2}") 2**

**Cevap 2**

**def YapDizi(n):**

**a = list() 2**

**for i in range(n): 3**

**a.append(random.randint(0, 999)) 3**

**return a 2**

**def YazDizi(a):**

**print("Dizi Elemanları") 2**

**for i in range(len(a)): 3**

**print(f"{i+1:4}.Eleman :{a[i]:5}") 3**

**def OkuTamsayı():**

**while True: 3**

**try: 2**

**return int(input("Tamsayı : ")) 3**

**except ValueError: 2**

**print("Geçerli Bir Tamsayı Giriniz\n") 2**

**def LinearSearch(a, x):**

**for i in range(len(a)): 3**

**if a[i] == x: 3**

**return i 2**

**return -1 2**

**Cevap 3**

**def YapDizi(n):**

**a = list() 2**

**for i in range(n): 3**

**a.append(random.randint(0, 999)) 3**

**return a 2**

**def YazDiziler(a, b, c):**

**print(" Dizi Artan Azalan") 2**

**print("---------------------------") 2**

**for i in range(len(a)): 3**

**print(f"{a[i]:5} {b[i]:8} {c[i]:9}") 3**

**def BubbleSort(a):**

**j = n = len(a) 3**

**while j > 0: 3**

**j = 0 2**

**for i in range(n-1): 3**

**if a[i] > a[i+1]: 3**

**a[i], a[i+1] = a[i+1], a[i] 4**

**j += 1 2**

**Cevap 4**

**def YapMatris(m, n):**

**if m > 0 and n > 0: 1.5**

**a = [n \* [0] for i in range(m)] 2.5**

**for i in range(m): 1.5**

**for j in range(n): 1.5**

**a[i][j] = 100\*random.random() – 50 2.5**

**else: 1.5**

**a = list() 1.5**

**return a 1.5**

**def YazMatrisEkran(a):**

**if len(a) < 1: 1.5**

**return 1.5**

**for i in range(len(a)): 1.5**

**for j in range(len(a[i])): 1.5**

**print(f"{a[i][j]:15.6f}", end="") 2**

**print() 1**

**def ÇarpMatris(a, b):**

**m, n, p = len(a), len(b[0]), len(b) 2.5**

**if m < 1 or n < 1 or p < 1: 1.5**

**return list() 1.5**

**c = [[0] \* n for i in range(m)] 2.5**

**for i in range(m): 1.5**

**for j in range(n): 1.5**

**for k in range(p): 1.5**

**c[i][j] += a[i][k] \* b[k][j] 3**

**return c 1.5**