

FİZ219_EST_UygulamaNotlari_11_AraSinav2

January 20, 2020

1 Uygulama Notları: 11

1.1 FİZ219 - Bilgisayar Programlama I | 20/01/2020

- 2. Ara Sınav Soru ve Çözümleri

Emre S. Tasci emre.tasci@hacettepe.edu.tr

1. Soru

```
[1]: function p = piramit(x)
    toplam=0;
    for i=1:x
        printf('\n');
        for j=1:i
            printf('%d ',j);
            toplam = toplam + j;
        endfor
    endfor
    p=toplam;
end
```

```
[2]: piramit(5)
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5 ans = 35
```

2. Soru

Aşağıdaki satırlarda önce yazım hatalarını işaretleyip, alttaki kutuya düzeltilmiş halini yazın; sonra da düzeltilmiş hali çalıştırdığınızda ekrana gelecek çıktıyı:

```
x'=-5; y=abs(x); z = 312.23456789 printf(x in degeri: 8.4%f - y nin degeri:
%2.1f z nin degeri: %6.2./n,x,y
```

```
[3]: x=-5; y=abs(x); z = 312.23456789;
```

```
printf("x in degeri: %8.4f - y nin degeri: %2.1f - z nin degeri: %6.2f.\n",x,y,z)
```

x in degeri: -5.0000 - y nin degeri: 5.0 - z nin degeri: 312.23.

3. Soru

$\vec{a} = 6\hat{i} - 2\hat{j} + 3\hat{k}$ ve $\vec{b} = 3\hat{i} + 4\hat{j} - 2\hat{k}$ vektörleri arasındaki açıyı (derece cinsinden) hesaplayan kod yazın.

İpucu: $\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos \theta$ ve aynı zamanda $\vec{a} \cdot \vec{b} = \sum_i a_i \cdot b_i$

```
[4]: a = [6 -2 3];  
b = [3 4 -2];  
boy_a = norm(a);  
boy_b = norm(b);  
a_dot_b = dot(a,b);  
cos_theta = a_dot_b / (boy_a * boy_b);  
theta = acosd(cos_theta)
```

theta = 83.909

```
[5]: # Munkun mertebe her seyi biz yapalim dersiniz:  
a = [6 -2 3];  
b = [3 4 -2];  
  
boy_a = 0;  
boy_b = 0;  
a_dot_b = 0;  
  
for i = 1:3  
    boy_a = boy_a + a(i)^2;  
    boy_b = boy_b + b(i)^2;  
    a_dot_b = a_dot_b + a(i)*b(i);  
endfor  
boy_a = sqrt(boy_a);  
boy_b = sqrt(boy_b);  
  
cos_theta = a_dot_b / (boy_a * boy_b);  
theta = acosd(cos_theta)
```

theta = 83.909

4. Soru

Bir dersin ağırlıklı yıl sonu ortalaması 2 ara sınavın notları ortalamasının %50'si, genel sınav notunun da %50'si alınıp, bunlar toplanarak hesaplanmaktadır. Parametre olarak bir öğrencinin 1. ara sınav, 2. ara sınav ve genel sınav notları verildiğinde, ağırlıklı yıl sonu ortalamasını hesaplayıp döndüren "ortalama" adında bir fonksiyon yazın.

```
[6]: # (tercihen) "ortalama.m" dosyasına
function ort = ortalama(s1,s2,s3)
ort = (s1+s2)/4 + s3/2;
endfunction

# Deneyelim:
ortalama(40,60,80)
```

ans = 65

Bonus: Fonksiyonumuza sadece bir öğrencinin notlarını değil, 150 kişilik bir sınıfın üç sınavının da notlarını içeren (150x3)lük bir “notlar” matrisi beslediğimizde ortalamaları (150x1)lik matris (/sütun vektörü) olarak döndürsün.

```
[7]: function ort = ortalama2(notlar)
ort = (notlar(:,1) + notlar(:,2))/4 + notlar(:,3)/2;
endfunction

# Deneyelim
notlar = randi([0 100],[150 3]);
notlar(1:5,:)

ortalamalar = ortalama2(notlar);
ortalamalar(1:5)
```

ans =

71	94	73
56	33	60
27	62	40
33	9	96
77	65	94

ans =

77.750
52.250
42.250
58.500
82.500

Soru 5 Aşağıdaki adımları yapan bir program yazın. Eğer yapamadığınız adım olursa, diğer adıma geçip, yapamadığınız adımın sonucunun belirtilen değişkende bulunduğunu varsayıp yolunuza o şekilde devam edebilirsiniz (ilk şıkta “m”; ikinci şıkta “k70” & “be70” dizileri).

i) [60,80) aralığında (yani) düzgün olasılık dağılımına uygun 100 tane rastgele ondalıklı sayı üretin, bunları m dizisinde/vektöründe toplayın (İpucu: rand() komutunu bu şekliyle çağırdığımızda [0,1) aralığında, düzgün olasılık dağılımına uygun bir sayı döndürdüğü bilgisiyle işe başlayabilirsiniz).

```
[8]: m = rand([1,100]) * 20 + 60;
```

```
# ilk 5'ini yazdiralim:
m(1:5)
# Araligi kontrol edelim:
min(m)
max(m)
```

```
ans =
```

```
71.011    69.698    66.001    61.433    72.091
```

```
ans = 60.047
```

```
ans = 79.920
```

ii) Sonrasında bunların 70'ten küçük olanlarını k70; 70'e eşit veya ondan büyük olanlarını da be70 adlarındaki iki dizide gruplayın.

```
[9]: k70 = m(m<70);
be70 = m(m>=70);
```

```
# Kontrol edelim:
min(k70)
max(k70)
min(be70)
max(be70)
```

```
ans = 60.047
```

```
ans = 69.985
```

```
ans = 70.390
```

```
ans = 79.920
```

iii) 70'ten küçük olanların kaçının [60,61) aralığında; kaçının [61,62) aralığında; ... kaçının [69,70) aralığında olduğunu saydırıp ekrana yazdırın

```
[10]: toplam = length(k70);
for ust_sinir=61:70
    alt_sinir = ust_sinir -1;
    ust_sinir_dan_kucuk_olanlar = sum(k70<ust_sinir);
    alt_sinir_dan_buyuk_esit_olanlar = sum(k70>=alt_sinir);
    ortak_olanlar = ust_sinir_dan_kucuk_olanlar +
    ↳alt_sinir_dan_buyuk_esit_olanlar ...
    - toplam;
    printf("[%d - %d) araliginda %d adet sayi vardır.\n",alt_sinir,...
    ust_sinir,ortak_olanlar)
endfor

# Bir araligi acip kontrol edelim:
k70(k70>=68 & k70<69)
```

```

[60 - 61) araliginda 8 adet sayi vardır.
[61 - 62) araliginda 4 adet sayi vardır.
[62 - 63) araliginda 5 adet sayi vardır.
[63 - 64) araliginda 6 adet sayi vardır.
[64 - 65) araliginda 6 adet sayi vardır.
[65 - 66) araliginda 3 adet sayi vardır.
[66 - 67) araliginda 6 adet sayi vardır.
[67 - 68) araliginda 6 adet sayi vardır.
[68 - 69) araliginda 4 adet sayi vardır.
[69 - 70) araliginda 9 adet sayi vardır.
ans =

```

```

68.804    68.552    68.583    68.214

```

```

[11]: # Alternatif hesap yontemi
for ust_sinir=61:70
    alt_sinir = ust_sinir -1;
    ust_sinir_dan_kucuk_olanlar = sum(k70<ust_sinir);
    alt_sinir_dan_kucuk_olanlar = sum(k70<alt_sinir);
    ortak_olanlar = ust_sinir_dan_kucuk_olanlar - alt_sinir_dan_kucuk_olanlar;
    printf("[%d - %d) araliginda %d adet sayi vardır.\n",alt_sinir,...
        ust_sinir,ortak_olanlar)
endfor

```

```

[60 - 61) araliginda 8 adet sayi vardır.
[61 - 62) araliginda 4 adet sayi vardır.
[62 - 63) araliginda 5 adet sayi vardır.
[63 - 64) araliginda 6 adet sayi vardır.
[64 - 65) araliginda 6 adet sayi vardır.
[65 - 66) araliginda 3 adet sayi vardır.
[66 - 67) araliginda 6 adet sayi vardır.
[67 - 68) araliginda 6 adet sayi vardır.
[68 - 69) araliginda 4 adet sayi vardır.
[69 - 70) araliginda 9 adet sayi vardır.

```

```

[12]: # Akillica bir yontem! (uzerinde calismanizi oneririm 8)
for ust_sinir=61:70
    alt_sinir = ust_sinir -1;
    ortak_olanlar = (k70<ust_sinir) + (k70>=alt_sinir);
    ortak_olanlar = sum (ortak_olanlar==2);
    printf("[%d - %d) araliginda %d adet sayi vardır.\n",alt_sinir,...
        ust_sinir,ortak_olanlar)
endfor

```

```

[60 - 61) araliginda 8 adet sayi vardır.
[61 - 62) araliginda 4 adet sayi vardır.
[62 - 63) araliginda 5 adet sayi vardır.

```

```

[63 - 64) araliginda 6 adet sayi vardır.
[64 - 65) araliginda 6 adet sayi vardır.
[65 - 66) araliginda 3 adet sayi vardır.
[66 - 67) araliginda 6 adet sayi vardır.
[67 - 68) araliginda 6 adet sayi vardır.
[68 - 69) araliginda 4 adet sayi vardır.
[69 - 70) araliginda 9 adet sayi vardır.

```

```

[13]: # Gormediginiz 'E' ("VE") operatoru kullanarak:
for ust_sinir=61:70
    alt_sinir = ust_sinir -1;
    ortak_olanlar = sum(k70<ust_sinir & k70>=alt_sinir);
    printf("[%d - %d) araliginda %d adet sayi vardır.\n",alt_sinir,...
        ust_sinir,ortak_olanlar)
endfor

```

```

[60 - 61) araliginda 8 adet sayi vardır.
[61 - 62) araliginda 4 adet sayi vardır.
[62 - 63) araliginda 5 adet sayi vardır.
[63 - 64) araliginda 6 adet sayi vardır.
[64 - 65) araliginda 6 adet sayi vardır.
[65 - 66) araliginda 3 adet sayi vardır.
[66 - 67) araliginda 6 adet sayi vardır.
[67 - 68) araliginda 6 adet sayi vardır.
[68 - 69) araliginda 4 adet sayi vardır.
[69 - 70) araliginda 9 adet sayi vardır.

```

Bonus Soru h_0 yüksekliğinden serbest düşmeye bırakılan bir top yere her çarpışından sonra bir önceki seferde ulaştığı en yüksek mesafenin yarısına kadar geri çıkabiliyor.

Bu topun verilen h_0 başlangıç yüksekliği için zamana göre yüksekliğini hesaplayıp çizen bir program yazın.

İpucu: $y(t) = y_0 + v_0 \cdot t + \frac{1}{2}at^2 \rightarrow y(t) = h_0 - \frac{1}{2}gt^2$

```

[14]: # (tercihen) "bouncy_y_t.m" dosyasına
function [y,t] = bouncy_y_t(h00,g=9.8,n_t=100)
# Calculates the altitude of a free falling object
# released from an initial height of h0, under
# gravitational acc. g, from t=0 to y=0 with
# the n_t time intervals
#
# Emre S. Tasci <emre.tasci@hacettepe.edu.tr>
# 26/12/2019

h0 = abs(h00);

t_f = sqrt(2*h0/g);
t = linspace(0,t_f,n_t);

```

```

y = h0-0.5*g*t.^2;

if(h00<0)
    t = [t t(end)+t];
    y1=flip(y);
    y = [y1 y];
endif

endfunction

```

```

[15]: %clear;

y = [];
t = [];

h0 = 120;
h00 = h0;
[yp,tp] = bouncy_y_t(h00,9.8,10);
[y] = [y yp];
[t] = [t tp];

for i=1:6
h00 /= 2;
end_tp = t(end);
[yp,tp] = bouncy_y_t(-h00,9.8,10);
[y] = [y yp];
[t] = [t end_tp+tp];
endfor

plot(t,y,"-ok");
yticks(0:10:h0);
ylim([0 h0]);
xlim([0 t(end)]);

```

