

MAT412 BİLGİSAYAR PROGRAMLAMA II FİNAL CEVAP ANAHTARI

S-1) Aşağıda verilmiş olan tablodan yararlanarak $x = 5$ noktasına karşılık gelen y değerini Linear Enterpolasyon kullanarak hesaplayan ve sonucu ekrana yedi basamak duyarlılıkla yazan bir kod yazınız.

x	4.0	4.5	5.5	6.0
y	1.3862944	1.5040774	1.7047481	1.7917595

C-1)

```
import math
x0=4.5
x1=5.5
x=5
y0=1.5040774
y1=1.7047481
p=(y1-y0)/(x1-x0)
y=y0+p*(x-x0)
print('% .7f ' % y)
```

S-2) $\int_2^4 \frac{x}{x-1} dx$ integralini aralığın 8 parçaya bölünmesi durumunda Simpson yöntemi ile hesaplayan ve sonucu ekrana sekiz basamak duyarlılıkla yazan bir kod yazınız.

C-2)

```
import numpy as np,math
f=lambda x:x/(x-1)
i=1
while i==1:
    n=input('aralık sayısını giriniz...');
    n=int(n)
    if n%2==0:
        a,b= input('sırasıyla a ve b değerlerini\'
        \' virgül ile ayırarak giriniz:').split(',');
        a,b=int(a),int(b)
        h=(b-a)/n
```

```

f=lambda x: x/(x-1)

t1=0

for i in np.arange(1,n,2):

    t1=t1+f(a+i*h)

t2=0

for j in np.arange(2,n-1,2):

    t2=t2+f(a+j*h)

print('integralin yaklaşık değeri:', '%.8f' % ((h/3)*(f(a)+f(b)+4*t1+2*t2)))

i=0

else:

    print('aralık sayısı çift olmalı!')

    print('aralık sayısını tekrar giriniz...')

```

S-3) $5 \sin^2 x - 8 \cos^5 x = 0$ denkleminin $[0.5,1.5]$ aralığındaki kökünü $x_0 = 0.5$ başlangıç yaklaşımı, $m = 11$ ardışık tekrar sayısı ve $\epsilon = 10^{-6}$ hata sınırı ile Newton yöntemine göre bulan ve sonucu ekrana 10 basamak duyarlılıkla yazan bir kod yazınız.

C-3)

```

import numpy as np,math

x=np.empty(20)

f=lambda x: 5*math.sin(x)*math.sin(x)-
8*math.cos(x)*math.cos(x)*math.cos(x)*math.cos(x)*math.cos(x)

g=lambda x: 10*math.sin(x)*math.cos(x)+30*math.cos(x)*math.sin(x)

x0,m, epsilon = input('sırasıyla x0,m ve epsilon değerlerini\'
' virgül ile ayırarak giriniz:').split(',');

x[0],m,epsilon=float(x[0]),int(m),float(epsilon)

print()

x[0]=x0

i=1

while (i<=m):

    x[i]=x[i-1]-f(x[i-1])/g(x[i-1])

    if abs(x[i]-x[i-1])<epsilon:

        print(i, '%.8f' % x[i], '%.10f' % x[i-1], '%.10f' % abs(x[i]-x[i-1]))

```

```

print()

print(i,'.ci iterasyonda kök değeri','%.10f'%x[i])

break

print(i,'%.8f'%x[i], '%.8f'%x[i-1], '%.10f'%abs(x[i]-x[i-1]))

i+=1

if i>m:

    print(i,'.ci iterasyonda kök bulunamadı')

```

S-4) $B = \begin{bmatrix} 1 & 5 & 1 \\ 2 & 3 & 1 \\ 3 & 1 & 4 \end{bmatrix}$ matrisinin

- a) Transpozunu(devriğini)
 - b) küpünü
 - c) sinüsünü
 - d) logaritmasını
- aynı program içinde hesaplayan bir kod yazınız
- C-4)**

```

import numpy as np,math
B=np.array([[1,5,1],[2,3,1],[3,1,4]])
TRANSPOZ=B.T
C=np.dot(B,B)
KUPU=np.dot(C,B)
SINUSU=np.sin(B)
LOGU=np.log(B)
print(TRANSPOZ)
print(KUPU)
print(SINUSU)
print(LOGU)

```

28.05.2020
Prof. Dr. Vedat Suat ERTÜRK