Scilab Textbook Companion for Principles of Electronics by V. K. Mehta and R. Mehta[1]

Created by
Kavan Patel
B.Tech
Instrumentation Engineering
Dharmsinh Desai University
College Teacher
None
Cross-Checked by
Chaitanya

July 11, 2017

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Principles of Electronics

Author: V. K. Mehta and R. Mehta

Publisher: S.Chand, New Delhi

Edition: 9

Year: 2005

ISBN: 81-219-2450-2

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Introduction

Scilab code Exa 1.1 terminal voltage

```
//chapter 1
//example 1.1
//page8

Eg=24 // V
Ri=.01 // ohm
P=100 // W

I=P/Eg // we know that P=Eg*I since for ideal source
    , V is equivalent to Eg
Vi=I*Ri
V=Eg-(I*Ri)

printf("voltage drop in internal resistance = %.3f V
    \n",Vi)
printf("terminal voltage = %.3f V",V)
```

Scilab code Exa 17.1 Example 1

```
//chapter17
//example17.1
//page375

L1=58.6d-6 // H
C1=300d-12 // F

f=1/(2*%pi*(L1*C1)^0.5)
printf("frequency of oscillations = %.3f Hz or %.3f
   kHz",f,f/1000)

// in book the answer is 1199 kHz but the accurate
   answer is 1200.358 kHz
```

Scilab code Exa 1.2 Exmaple 2

```
//chapter1
//example1.2
//page10

Eg=500 // V
Ri=1000 // ohm

// for Rl=10 ohm
Rl1=10 // ohm
I1=Eg/(Rl1+Ri)
printf("load current for Rl=10ohm is %.3f A \n",I1)

// for Rl=10 ohm
Rl2=50 // ohm
I2=Eg/(Rl2+Ri)
printf("load current for Rl=50ohm is %.3f A \n",I2)

// for Rl=10 ohm
Rl3=100 // ohm
```

```
I3=Eg/(Rl3+Ri)
printf("load current for Rl=100ohm is %.3f A",I3)
```

Scilab code Exa 1.3 Exmaple 3

```
//chapter1
//example1.3
//page11

V=10 // V
R=10 // ohm

I=V/R // calculate short-circuit current by shorting
    AB
printf("equivalent current source has magnitude = %
   .3f A",I)

// no load is connected across AB and 10V source has
     negligible resistance
// so resistance across AB is 10 ohm

// the constant voltage source when converted to
   constant current source will thus have a source
   of 1A in parallel with resistor of 10 ohm
```

Scilab code Exa 1.4 Exmaple 4

```
//chapter1
//example1.4
//page12

I=6 // mA
R=2 // kilo ohm
```

```
V=I*R // by ohm law
printf("voltage of voltage source = %.3f V",V)

// this voltage source when connected in series with
    2000 ohm gives equivalent voltage source for the
    given constant current source
```

Scilab code Exa 1.5 Exmaple 5

```
//chapter1
//example1.5
//page13

E=200 // V
Ri=100 // ohm

Rl=100 // for load=100ohm
I=E/(Ri+Rl)
Pl=I^2*Rl
Pt=I^2*(Rl+Ri)
efficiency=(Pl/Pt)*100
printf("for load=100 ohm, power delivered to load= %
   .3f W and efficiency=%.3f percentage \n \n",Pl,
   efficiency)

Rl=300 //for load=300ohm
I=E/(Ri+Rl)
Pl=I^2*Rl
Pt=I^2*(Rl+Ri)
efficiency=(Pl/Pt)*100
printf("for load=300 ohm, power delivered to load= %
   .3f W and efficiency=%.3f percentage \n \n",Pl,
   efficiency)

```

```
printf("comment: \n ")
printf("if load resistance is equal to internal resistance ,maximum power is \n transferred but efficiency is low \n ")
printf("if load resistance is more than internal resistance , power transferred \n is less but efficiency is high")
```

Scilab code Exa 1.6 Exmaple 6

```
//chapter1
//example1.6
//page14

//for maximum power transfer , resistance of load and amplifier should match
//so we take load=15 ohm

Rl=15 // ohm
Ri=15 // ohm
V=12 // V

Rt=Rl+Ri
I=V/Rt
P=I^2*Rl

printf("for maximum power transfer load must equal amplifier resistance \nso required load = %d ohm\n \n",Rl)
printf("power delivered to load = %.3f W",P)
```

Scilab code Exa 1.7 Exmaple 7

```
//chapter1
//example1.7
//page14

V=50 // V
Rl=100 // ohm
Zi=100+50*%i
//for maximum power transfer load impedence should
   be conjugate of internal resistance so
Zl=100-50*%i

Zt=Zi+Zl
I=V/Zt
P=I^2*Rl

printf("load for maximum power (in ohms)=")
disp(Zl)

printf("maximum power transfered to load=%.3f W",P)
```

Scilab code Exa 1.8 Exmaple 8

```
//chapter1
//example1.8
//page21

R=8 // ohm
R1=10 // ohm
R2=20 // ohm
R3=12 // ohm
//removing 100 ohm resistance , we form linear
   equations by assuming currents I1 through loop1
   and I2 through loop2

//100=10*I1+20*(I1-I2)
```

```
//0=(12+8)*I2+20*(I2-I1)

//thus we get the following linear equations

//30*I1-20*I2=100
//-20*I1+40*I2=0
//solving these equations

a=[30 -20;-20 40]
b=[100;0]
x=inv(a)*b // matrix of I1 and I2

I2=x(2,1) // current through 8 ohm resistor

E0=I2*R
printf("voltage across AB with 100 ohm resistance
    not connected = %.3f V \n",E0)

R_equi=(R1*R2/(R1+R2))+R3
R0=R_equi*R/(R_equi+R)
printf("resistance between AB with 100 ohm removed
    and voltage source shorted = %.3f ohm \n",R0)

I=E0/(R0+R1)
printf("current through 100 ohm resistor = %.3f A",I
    )
```

Scilab code Exa 1.9 Exmaple 9

```
//chapter1
//exzmple1.8
//page16

R1=1 // kilo ohm
R2=1 // kilo ohm
```

```
R3=1 // kilo ohm
V=20 // V

E0=(R3/(R1+R2))*V // thevenin voltage = voltage
   across R3 since A and B are open circuited which
   means no drop across R2
R0=R2+(R1*R3/(R1+R3)) // thevenin resistance =
   resistance between A and B with no load and
   voltage source shorted

printf("thevenin voltage = %.2f V \nthevenin
   resistance = %.2f kilo ohm",E0,R0)
```

Scilab code Exa 1.10 Exmaple 10

```
//chapter1
//example1.10
//page18

V=120 // V
R1=40 // ohm
R2=20 // ohm
R3=60 // ohm

//removing load, voltage across AB is
E0=R2*V/(R1+R2)

//replacing voltage source by short and removing
   load, resistance across AB is
R0=R3+(R1*R2/(R1+R2))

//for maximum power transfer, load must be equal to
   resistance across AB so
Rl=R0

```

```
P=E0^2/(4*Rl)
printf("load resistance for maximum power transfer =
    %.3f ohm \n",Rl)
printf("maximum power to load = %.3f W",P)
```

Scilab code Exa 1.11 Exmaple 11

```
//chapter1
//example1.11
//page20

R1=4 // ohm
R2=6 // ohm
R3=5 // ohm
R4=8 // ohm
V=40 // V

// load is removed and A and B are shorted
load_source=R1+(R2*R3/(R2+R3))
source_current=V/load_source

norton_current=source_current*(R2/(R2+R3)) // short
    circuit current in AB

printf("shortcircuit current in AB = %.3f A \n",
    norton_current)

// load is removed and battery is replaced by a
    short
norton_resistance=R3+(R1*R2/(R1+R2))
printf("norton resistance= %.3f ohm \n",
    norton_resistance)

// equivalent circuit is norton current source in
    parallel with norton resistance
```

```
I=norton_current*(norton_resistance/(
    norton_resistance+R4)) // current through 8 ohm
    resistance
printf("current through 8ohm resistor = %.3f A",I)
```

Scilab code Exa 1.12 Exmaple 12

```
// chapter 1
// example 1.12
// page 21

printf("To find Norton equivalent circuit we need to
     find \nNorton current I_N and Norton resistance
    R_N \n \n")
printf("If Thevenin resistnce = Ro and Thevenin
    voltage = Eo then \n \n")
printf("To convert Thevenin circuit to Norton
    circuit, \n")
printf("I_N=Eo/Ro and R_N=Ro \n \n")
printf("To convert Norton circuit to Thevenin
    circuit, \n")
printf("Eo=I_N*R_N and Ro=R_N \n")
```

Chapter 2

Electron emission

Scilab code Exa 2.1 Example 1

```
//chapter 2
//example 2.1
//page 29

A=60.2d4 // ampere per square m per square kelvin
T=2500 // kelvin
phi=4.517 // eV
d=0.01d-2 // m
l=5d-2 // m

b=11600*phi
Js=A*T^2*exp(-b/T)
a=%pi*d*l

emission_current=Js*a

printf("emission current=%f A",emission_current)
```

Scilab code Exa 2.2 Example 2

```
// Chapter 2
// example 2.2
// page 29
Js=0.1 // ampere per square cm
A=60.2 // ampere per square cm per square kelvin
T=1900 // kelvin

// Js=A*T^2*exp(-b/T) so b=-T*log(Js/(A*T^2))

b=-T*log(Js/(A*T^2))

// b=11600*phi so making phi as subject

phi=b/11600

printf("work function=%f eV \n",phi)
// the accurate answer is 3.521466
// but in the book it is mistakenly written as 3.56

if(2.63<phi & phi<4.52)
    printf("thoriated tungsten has work function between 2.63eV to 4.52eV.\nSo sample is likely to be thoriated tingsten")
elseif(phi<=2.63 | phi>4.52)
        printf("tungsten is contaminated") // for pure tungsten, phi must be 4.52 exactly
else
    printf("tungsten is pure") // phi=4.52 implies tungsten is pure
end

// please note that there is error in the answer of work function phi in the book
// The correct answer is 3.521466 eV and not 3.56 eV
```

Chapter 3

Vacuum tubes

Scilab code Exa 3.1 Example 1

```
//chapter3
//example3.1
//page41

Ib1=10 // mA
Eb1=100 // V
Ib2=20 // mA

// Ib is proportional to Eb^(3/2)
// so we can say Ib1/Ib2 = Eb1^1.5/Eb2^1.5
//thus we can write

log_Eb2=(2/3)*log(Eb1^1.5*Ib2/Ib1)
Eb2=exp(log_Eb2)
printf("required plate voltage = %.3f V",Eb2)
```

Scilab code Exa 3.2 Example 2

```
//chapter3
//example3.2
//page49

mu=20
rp=8000 // ohm

gm=mu/rp // since mu=rp*gm
gm_micro=gm*10^6 //micro mho
printf("mutual conductance of triode = %f mho or %.3
    f micro mho",gm,gm_micro)
```

Scilab code Exa 3.3 Example 3

```
//chapter3
//example3.3
//page49

// for constant Ec=-1.5
Eb1=100 // V
Eb2=150 // V
Ib1=7.5d-3 // A
Ib2=12d-3 // A

Eb_diff=Eb2-Eb1
Ib_diff=Ib2-Ib1

rp=Eb_diff/Ib_diff
rp_kilo_ohm=rp/10^3 //kilo ohm

printf("plate resistance = %.3f ohm or %.3f kilo ohm
    \n",rp,rp_kilo_ohm)

// for constant Eb=150
Ib1=5d-3 // A
```

```
Ib2=12d-3 // A
Ec1=-3 // V
Ec2=-1.5 // v

Ib_diff=Ib2-Ib1
Ec_diff=Ec2-Ec1

gm=Ib_diff/Ec_diff
gm_micro_mho=gm*10^6 //micro mho
printf("mutual conductance=%.3f mho or %.3f micro
   mho \n",gm,gm_micro_mho)

mu=rp*gm
printf("amplification factor = %.3f",mu)

//in book the answer of amplification factor i.e.
   51.852 is rounded off to 52
```

Scilab code Exa 3.4 Example 4

```
//chapter3
//example3.4
//page50

Eb=250 // V
Ec=-3 // V

// given that Ib=0.003*(Eb+30*Ec)^1.5 mA
// differentiating w.r.t Ec with Eb=constant, we get
gm=0.003*1.5*(Eb+30*Ec)^0.5*30*10^-3
mutual_inductance_micro=gm*10^6

printf("mutual conductance = %f mho or %.3f micro
   mho \n",gm,mutual_inductance_micro)

```

```
// differentiating given equation w.r.t Ec with Ib=constant , we get
// 0=0.003*10^-3*1.5*(Eb+Ec)^1.5*(mu+30) where mu is equal to ratio of changes in Eb and Ec i.e. amplification factor
// thus mu+30=0 hence we get
mu=-30
    printf("here negative sign of amplification factor indicates that Eb and Ec are in opposite direction. \n \n")
// here we need not worry as to if mu may be positive because the equation given in problem statement will always give mu+30=0 i.e. mu=-30

printf("amplification factor = %.3f \n",mu)

rp=mu/gm
if rp<0    // rp can not be negative
    rp=-rp
end

printf("plate resistance = %.3f ohm \n",rp)

//in book, the answers are less accurate. The accurate answers are
// gm=1707.630 micro mho
// plate resistance=17568.209 ohm
```

Scilab code Exa 3.5 Example 5

```
//chapter3
//example3.5
//page58

// use of Rsg = to obtain desired potential on
```

```
   screen grid since it is connected between power
   supply and screen grid
// use of Csg = to provide ac grounding for the
   screen

Ebb=300 // V
Ib=10d-3 // A
Rl=4.7d3 // ohm
Rk=68 // ohm
Isg=3d-3 // A
Vsg=150 // V

cathode_voltage=Ebb-(Ib*Rl)
grid_cathode_bias=-Rk*(Ib+Isg) // since current
   through cathode resistance is Ib+Isg
Rsg=(Ebb-Vsg)/Isg // since plate supply voltage =
   grid voltage + drop across Rsg
Rsg_kilo_ohm=Rsg/10^3 // in kilo ohm

printf("zero signal plate cathode voltage = %.3f V \
   n",cathode_voltage)
printf("grid cathode bias = %.3f V \n",
   grid_cathode_bias)
printf("Resistor Rsg = %.3f ohm or %.3f kilo ohm \n"
   ,Rsg,Rsg_kilo_ohm)
```

Chapter 4

Vacuum tube rectifiers

Scilab code Exa 4.1 Example 1

```
//chapter4
//example4.1
//page68

rp=300 // ohm
Rl=1200 // ohm

Vm=200*2^0.5 //V
Im=Vm/(rp+Rl)
Idc=Im/%pi // in ampere
Idc_mA=Idc*1000 // in mA
Irms=Im/2
Irms_mA=Irms*1000
Pdc=Idc^2*Rl
Pac=Irms^2*(rp+Rl)
efficiency=(Pdc/Pac)*100

printf("dc current = %.3f A or %.3f mA \n",Idc,
    Idc_mA)
printf("rms current = %.3f A or %.3f mA \n",Irms,
    Irms_mA)
```

```
printf("rectification efficiency = %.2f percentage",
    efficiency)

// accurate answer of rms current is 94.281 mA but
    in book it is given as 94.5 mA
```

Scilab code Exa 4.2 Example 2

```
//chapter4
//example4.2
//page68

rp=200 // ohm
Rl=800 // ohm
Edc=100 // V

// if maximum ac voltage required=Vm then
// Edc=Idc*Rl i.e. Edc=Vm*Rl/(%pi*(rp+Rl))
// thus

Vm=Edc*%pi*(rp+Rl)/Rl
efficiency=(0.406/(1+(rp/Rl)))*100

printf("required ac voltage = %.3f V \n",Vm)
printf("rectification efficiency = %.3f percentage",
    efficiency)
```

Scilab code Exa 4.3 Example 3

```
//chapter4
//example4.3
//page69

```

```
Vm=1000 // V
rp=500 // ohm
Rl=4500 // ohm

Im=Vm/(rp+Rl) // in A
Idc=Im/%pi // in A
Idc_mA=Idc*1000 // in mA
Irms=Im/2 // since ac current is equal to rms
    current
Irms_mA=Irms*1000 // in mA
W=Irms^2*(rp+Rl) // in watts

printf("dc ammeter reading = %.3f A or %.3f mA \n",
    Idc,Idc_mA)
printf("reading of ac ammeter = %.3f A or %.3f mA \n
    ",Irms,Irms_mA)
printf("reading of wattmeter = %.3f W",W)
```

Scilab code Exa 17.4 Example 4

```
//chapter17
//example17.4
//page381

R=1d6 // ohm
C=68d-12 // F

fo=1/(2*%pi*R*C*(6)^0.5)
printf("frequency of oscillations = %.3f Hz",fo)

// in book the answer given is 954 Hz but the
    accurate answer is 955.511 Hz
```

Scilab code Exa 4.4 Example 4

```
//chapter4
//example4.4
//page74

Vs=300 // V
rp=500 // ohm
Rl=2000 // ohm
Vm=Vs*2^0.5 // in V
Im=Vm/(rp+Rl) // A
Idc=2*Im/%pi // A
Pdc=Idc^2*Rl // W
Irms=Im/2^0.5 //A
Pac=Irms^2*(rp+Rl) // W
efficiency=(Pdc/Pac)*100

printf("dc power output = %.3f W \n",Pdc)
printf("ac power input = %.3f W \n",Pac)
printf("efficiency = %.2f percentage",efficiency)
```

Scilab code Exa 4.5 Example 5

```
//chapter4
//example4.5
//page74

Vm=1000 // V
rp=500 // ohm
Rl=4500 // ohm

Im=Vm/(rp+Rl) // in ampere
Idc=2*Im/%pi // in ampere
Idc_mA=Idc*1000 // in mA
Iac=Im/2^0.5 // in ampere
```

```
Iac_mA=Iac*1000 // in mA

printf("dc ammeter reading = %.3f A or %.3f mA \n",
   Idc,Idc_mA)
printf("ac ammeter reading = %.3f A or %.3f mA",Iac,
   Iac_mA)
```

Chapter 5

Vacuum tube amplifiers

Scilab code Exa 5.1 Example 1

```
//chspter5
//example5.1
//page85

mu=20
rp=10 // kilo ohm
Rl=15 // kilo ohm

Av=mu*Rl/(rp+Rl)

printf("voltage gain = %.3f",Av)
```

Scilab code Exa 5.2 Example 2

```
//chspter5
//example5.2
//page85

```

```
mu=20
rp=10 // kilo ohm
Rl=15 // kilo ohm
Eg=3 // V

// the diagram in book is for understanding only.
   Also we do not have a block of "triode" in scilab
    xcos. The figure is not required to solve the
   problem.

Av=mu*Rl/(rp+Rl)
Ip=(mu*Eg/2^0.5)/(rp+Rl)
V_out=Ip*Rl

printf("voltage gain = %.3f \n",Av)
printf("load current = %.3f mA \n",Ip)
printf("output voltage = %.3f V",V_out)

// the accurate answer for output voltage is 25.456V
    but in book it is given as 25.35V
```

Scilab code Exa 5.3 Example 3

```
//chapter5
//example5.3
//page85

// for Rl=50, Av=30
//for Rl=85, Av=34

// Av=mu*Rl/(rp+Rl)
// thus
// Av*rp-mu*Rl=-Av*rl
// substituting for Rl=50 and Rl=85 we get the
   following linaer equations
```

```
// 30*rp-50*mu=-1500 and
// 34*rp-85*mu=-2890
// solving by matrix

a=[30 34 ; -50 -85]
b=[-1500 -2890]
solution=b/a
mu=solution(1,2)
rp=solution(1,1) // in kilo ohms since RL was in
   kilo ohm in the equations

gm_kilo_mho=mu/rp
gm=gm_kilo_mho/1000
printf("mu = %.3f \n",mu)
printf("rp = %.3f kilo ohm \n",rp)
printf("gm = %.4f mho \n",gm)
```

Scilab code Exa 5.4 Example 4

```
//chapter5
//example5.4
//page86

mu=6
Eg=9 // V
rp=2400 // ohm
Rl=3000 // ohm

Ip=mu*Eg/(rp+Rl) // A
power=Ip^2*Rl // W

printf("ac power in load = %.3f W",power)
```

Scilab code Exa 5.5 Example 5

```
//chapter5
//example5.5
//page95

rp=1000 // ohm
Rl=10 // ohm
Eg=8 // V
mu=20

// the diagram in book is for understanding only.
    Also we do not have a block of "triode" in scilab
     xcos. The figure is not required to solve the
    problem.
// however, the equivalent circuit has been drawn in
     xcos for reference.

// since rp=n^2*Rl for maximum power transfer so
n=(rp/Rl)^0.5

// P_max=Ip^2*RE where Ip=mu*Eg/(rp+RE) and RE=rp
// thus
P_max=((mu*Eg)^2)/(4*rp)

printf("transformation ratio n= %.2f \n",n)
printf("power supplied to speaker when signal is 8V
    rms is = %.3f W",P_max)
```

Chapter 9

Semiconductor diode

Scilab code Exa 9.1 Example 1

```
//chapter9
//example9.1
//page142

printf("in fig. (i), the conventional current coming out of battery flows in the \nbranch circuits. In diode D1,the conventional current flows in the \ndirection of arrowhead and hence this diode is forward biased. \nHowever in diode D2, the conventional current flows opposite \nto arrowhead and hence this diode is reverse biased.\n \n")
printf("in fig. (ii), During the positive half cycle of input ac voltage, the \nconventional current flows in the direction of arrowhead and hence diode \nis forward biased. However, during the negative half cycle \nof input ac voltage, the diode is reverse biased.\n \n")
printf("in fig. (iii), During the positive half cycle of input ac voltage, the \nconventional current flows in the direction of arrowhead in D1
```

```
    but it flows \nopposite to arrowhead in D2. So
    during positive half cycle , \ndiode D1 is forward
    biased and diode D2 is reverse biased. \nHowever
    in the negative half cycle of the input ac
    voltage , diode D2 \nis forward biased and diode
    D1 is reverse biased.\n \n")
printf("in fig. (iv),  During the positive half
    cycle of input ac voltage , \nboth diodes are
    reverse biased. However in the negative half
    cycle of the \ninput ac voltage , both diodes are
    forward biased.\n \n")
```

Scilab code Exa 9.2 Example 2

```
//chapter9
//example9.2
//page145

Vi_p=20 // V
rf=10 // ohm
Rl=500 // ohm
Vo=0.7 // V
Vin=20 // V

// peak current through diode will occur when Vin=Vf
    so
Vf=Vin
// since Vf=Vo+If_peak(rf+Rl) making If_peak as
    subject we get
If_peak1=(Vf-Vo)/(rf+Rl) // in ampere
Vout_peak1=If_peak1*Rl

// for ideal diode, Vo=0 and rf=0 so
// Vf=If_peak*Rl so we get
If_peak2=Vf/Rl  // in ampere
```

```
Vout_peak2=If_peak2*Rl

printf("peak current through given diode = %.3f mA
   and peak output voltage = %.3f V \n",If_peak1
   *1000,Vout_peak1)
printf("peak current through ideal diode = %.3f mA
   and peak output voltage = %.3f V \n",If_peak2
   *1000,Vout_peak2)
```

Scilab code Exa 9.3 Example 3

```
//chapter9
//example9.3
//page146

R1=50 // ohm
R2=5 // ohm
V=10 // V

Eo=V*R2/(R1+R2) // thevenin voltage
Ro=R1*R2/(R1+R2) // thevenin resistance
I_D=Eo/Ro // current through diode in ampere

printf("current through diode = %.3f mA \n",I_D
   *1000)
```

Scilab code Exa 9.4 Example 4

```
//chapter9
//example9.4
//page146

V=10 // V
```

```
V_D1=0.7 // V
V_D2=0.7 // V
R=48 // ohm
R_D1=1 // ohm
R_D2=1 // ohm

// D1 and D3 are forward biased while D2 and D4 are
   reverse biased thus
V_net=V-V_D1-V_D2
R_t=R_D1+R+R_D2
I=V_net/R_t

printf("circuit current = %.3f mA \n",I*1000)
```

Scilab code Exa 9.5 Example 5

```
//chapter9
//example9.5
//page147

E1=24 // V
E2=4 // V
Vo=0.7 // V
R=2 // kilo ohm

// diode D1 is forward biased and diode D2 is
   reverse biased so
I=(E1-E2-Vo)/R

printf("current in the circuit = %.3f mA \n",I)
```

Scilab code Exa 9.6 Example 6

```
//chapter9
//example9.6
//page147

V=20 // V
V_D_Ge=0.3 // V

// when voltage is applied, Ge diode turns on first
    and 0.3 V is maintained across circuit so Si
    diode never turns on. So
V_A=V-V_D_Ge

printf("voltage V_A at point A = %.3f V \n",V_A)
```

Scilab code Exa 9.7 Example 7

```
//chapter9
//example9.7
//page148

V=10 // V
V_D=0.7 // V
R_BC=2 // kilo ohm
R=2 // kilo ohm

// by Kirchoff voltage law we get
// -V_D-I_D*R_BC-2*I_D*R+V=0 thus making I_D as
    subject we get
I_D=(V-V_D)/(R_BC+2*R)
V_Q=2*I_D*R

printf("I_D = %.3f mA \n",I_D)
printf("V_Q = %.3f V \n",V_Q)
```

Scilab code Exa 9.8 Example 8

```
//chapter9
//example9.8
//page148

V=15 // V
R=0.5 // kilo ohm
V_D=0.7 // V

// both diodes are forward biased

I1=(V-V_D)/R
I_D1=I1/2
I_D2=I_D1

printf("current through diode D1 = %.3f mA and diode D2 = %.3f mA \n",I_D1,I_D2)
```

Scilab code Exa 9.9 Example 9

```
//chapter9
//example9.9
//page151

P_dc=40 // W
P_ac=100 // W

efficiency=100*P_dc/P_ac

printf("rectification efficiency = %.3f percent \n \n",efficiency)
```

```
printf("remaining 60 watts are not lost. Crystal diode consumes only a \nlittle power due to its small internal resistance. \nActualy 100 W ac power is contained as 50 W in positive half \ncycle and 50 W in negative half cycle.\nThe 50 W of negative half cycle are not supplied at all. \nThe 50 W of positive half cycle are converted to 40 W \n")
```

Scilab code Exa 9.10 Example 10

```
//chapter9
//example9.10
//page152

n=10
Vp=230 // V

Vpm=2^0.5*Vp
Vsm=Vpm/n // since n=Vpm/Vsm=N1/N2

// Idc=Im/%pi and Vdc=Idc*Rl so
// Vdc=(Im/%pi)*Rl .Also Im*Rl=Vsm so
Vdc=Vsm/%pi

// in negative half cycle diode is reverse biased so maximum secondary voltage appears across diode.
PIV=Vsm

printf("output dc voltage = %.2f V \n",Vdc)
printf("peak inverse voltage = %.2f V \n",PIV)

// accurate answer for output dc voltage is 10.35 V not 10.36 V
```

Scilab code Exa 9.11 Example 11

```
//chapter9
//example9.11
//page152

rf=20 // ohm
Rl=800 // ohm
Vm=50 // V

Im=Vm/(rf+Rl) // in ampere
Idc=Im/%pi // in ampere
Irms=Im/2 // in ampere
Pac=Irms^2*(rf+Rl)
Pdc=Idc^2*Rl
Vout=Idc*Rl
efficiency=100*Pdc/Pac

printf("Im = %.1f mA \n",Im*1000)
printf("Idc = %.1f mA \n",Idc*1000)
printf("Irms = %.1f mA \n \n",Irms*1000)
printf("ac power input = %.3f W \n",Pac)
printf("dc power output = %.3f W \n \n",Pdc)
printf("dc output voltage = %.3f V \n \n",Vout)
printf("efficiency = %.3f percent \n",efficiency)
```

Scilab code Exa 9.12 Example 12

```
//chapter9
//example9.12
//page153

```

```
Vdc=50 // V
rf=25 // ohm
Rl=800 // ohm

// Vdc=Idc*Rl and Idc=Im/%pi so
// Vdc=Im*Rl/%pi
// but Im=Vm/(rf+Rl) so
// Vdc=Vm*Rl/(%pi*(rf+Rl))
// making Vm as subject we get

Vm=Vdc*%pi*(rf+Rl)/Rl

printf("ac voltage required = %.1f V \n",Vm)
```

Scilab code Exa 9.13 Example 13

```
//chapter9
//example9.13
//page157

rf=20 // ohm
Rl=980 // ohm
Vs=50 // V

Vm=Vs*2^0.5
Im=Vm/(rf+Rl)
Idc=2*Im/%pi // in ampere
Irms=Im/2^0.5 // in ampere

printf("mean load current  = %.3f mA \n",Idc*1000)
printf("rms load current = %.3f mA \n",Irms*1000)
```

Scilab code Exa 19.14 power

```
//chapter19
//example19.14
//page424

Vc=5 // V
V_lower=2.5 // V
V_upper=2.5 // V
R=2 // kilo ohm

// figure given in book is just for understanding
    purpose.It is not a part of solution.
// however, the figure has been made in xcos and
    screenshot has been attached for reference

// since power=(rms voltage)^2/R we get

Pc=(0.707*Vc)^2/R
P_lower=(0.707*V_lower)^2/R
P_upper=(0.707*V_upper)^2/R
Pt=Pc+P_lower+P_upper

printf("power delivered by carrier = %.3f mW \n",Pc)
printf("power delivered by lower sideband = %.3f mW
    \n",P_lower)
printf("power delivered by upper sideband = %.3f mW
    \n",P_upper)
printf("total power delivered by AM wave = %.3f mW \
    n",Pt)
```

Scilab code Exa 9.14 Example 14

```
//chapter9
//example9.14
//page157

```

```
rf=0
n=5
Vp=230 // V rms
Rl=100 //ohm

Vs=Vp/n // V rms
Vsm=Vs*2^0.5 // maximum voltage across secondary
Vm=Vsm/2 // maximum voltage across half secondary
    winding

Idc=2*Vm/(%pi*Rl)
Vdc=Idc*Rl
PIV=Vsm
efficiency=100*0.812/(1+rf/Rl)

printf("dc output voltage = %.3f V \n",Vdc)
printf("PIV = %.3f V \n",PIV)
printf("efficiency = %.3f percent \n",efficiency)
```

Scilab code Exa 9.15 Example 15

```
//chapter9
//example9.15
//page158

n=4
Rl=200 // ohm
fin=50 // Hz
Vp=230 // V rms

Vs=Vp/n // V rms
Vsm=Vs*2^0.5 // maximum voltage across secondary

Idc=2*Vsm/(%pi*Rl)
Vdc=Idc*Rl
```

```
PIV=Vsm

// in full wave rectifier , output frequency is twice
    input frequency since there are two ouput pulses
    for each cycle of input
fout=2*fin

printf("dc output voltage = %.3f V \n",Vdc)
printf("peak inverse voltage = %.3f V \n",PIV)
printf("output frequency = %.3f Hz",fout)

// the accurate answer for dc output voltage is
   51.768 V but in book it is given as 52 V
```

Scilab code Exa 9.16 Example 16

```
//chapter9
//example9.16
//page158

// for dc output
    // for centre-tap circuit
        n=5
        Vp=230 // V rms
        Rl=100 //ohm
        Vs=Vp/n // V rms
        Vsm=Vs*2^0.5 // maximum voltage across
           secondary
        Vm=Vsm/2 // maximum voltage across half
           secondary winding
        Vdc=2*Vm/%pi // since Vdc=Idc*Rl and Idc=2*
           Vm/(%pi*Rl)

    // for bridge circuit
        n_dash=5
```

```
        Vp_dash=230 // V rms
        Rl_dash=100 //ohm
        Vs_dash=Vp_dash/n_dash// V rms
        Vsm_dash=Vs*2^0.5 // maximum voltage across secondary
        Vm_dash=Vsm_dash
        Vdc_dash=2*Vm_dash/%pi // since Vdc=Idc*Rl and Idc=2*Vm/(%pi*Rl)


// for same dc output Vm must be same for both circuits i.e. n=5 for centre-tap and n=10 for bridge
    // for centre-tap circuit
        n1=5
        Vs1=Vp/n1 // V rms
        Vsm1=Vs1*2^0.5 // maximum voltage across secondary
        Vm1=Vsm1/2
        PIV1=2*Vm1

    // for bridge circuit
        n2=5
        Vs2=Vp/n2 // V rms
        Vsm2=Vs2*2^0.5 // maximum voltage across secondary
        Vm2=Vsm2/2
        PIV2=Vm2

printf("dc output voltage for centre-tap circuit = %.3f V \n",Vdc)
printf("dc output voltage for bridge circuit = %.3f V \n \n",Vdc_dash)

printf("for same output, PIV for centre-tap circuit = %.3f V and bridge circuit = %.3f V \n",PIV1,PIV2)

```

```
// the figure of transformer is for reference only.
   Also it cannot be plotted in scilab since scilab
   does not have centre-tap transformer
```

Scilab code Exa 9.17 Example 17

```
//chapter9
//example9.17
//page160

Vin=240 // V rms
Rl=480 // ohm
rf=1 // ohm

Vm=Vin*2^0.5
// for bridge rectifier we know that
Im=Vm/(2*rf+Rl)
Idc=2*Im/%pi
Irms=Im/2
P=Irms^2*rf

printf("mean load current = %.3f A \n",Idc)
printf("power dissipated in each diode = %.3f W \n",
   P)

// the accurate answers are mean load current =
   0.448 A and power dissipated in each diode =
   0.124 W
```

Scilab code Exa 9.18 Example 18

```
//chapter9
//example9.18
```

```
//page162

Vrms_A=0.5 // V
Vdc_A=10 // V
Vrms_B=1 // V
Vdc_B=25 // V

ripple_A=Vrms_A/Vdc_A
ripple_B=Vrms_B/Vdc_B

if ripple_A>ripple_B
    printf("power supply B is better \n")
elseif ripple_B>ripple_A
   printf("power supply A is better \n")
else
    printf("both are equal \n")
end
```

Scilab code Exa 9.19 Example 19

```
//chapter9
//example9.19
//page165

// the waveform given in book is for understanding
   only. It is not required to solve the problem.
   Also it cannot be plotted in scilab unless Vm and
    Vdc are given.

R=25 // ohm
Rl=750 // ohm
Vm=25.7 // V

Vdc_dash=2*Vm/%pi
Vdc=Vdc_dash*Rl/(R+Rl)
```

```
printf("voltage across load is %.3f V plus a small
    ripple \n",Vdc)

// the accurate answer is 15.833 V but in book it is
     given as 15.9 V
```

Scilab code Exa 9.20 Example 20

```
//chapter9
//example9.20
//page170

R=5 // kilo ohm
Rl=10 // kilo ohm
Ei=120 // V
Vz=50 // V

V=Ei*Rl/(R+Rl) // voltage across open circuit if
    zener diode is removed
Vo=Vz // output voltage
V_R=Ei-Vz // drop across R
Il=Vz/Rl // load current
I=V_R/R // current through R

// by Kirchoff first law I=Iz+Il
Iz=I-Il

printf("output voltage = %.3f V \n",Vo)
printf("voltage drop across series resistance = %.3f
    V \n",V_R)
printf("current through Zener diode = %.3f mA \n",Iz
    )
```

Scilab code Exa 9.21 Example 21

```
//chapter9
//example9.21
//page171

Vmax=120 // V
Vmin=80 // V
Vz=50 // V
R_L=10 // kilo ohm
R1=5 // kilo ohm

// zener diode is on for Vmax and Vmin both since
   they are > Vz

// for max Iz
    V_R1=Vmax-Vz
    I=V_R1/R1 // current through R1
    I_L=Vz/R_L // current through load
    // by Kirchoff first law I=I_L+Iz so applying it
        we get
    Iz_max=I-I_L

// for min Iz
    V_R1_dash=Vmin-Vz
    I_dash=V_R1_dash/R1// current through R1
    I_L_dash=Vz/R_L // current through load
    // by Kirchoff first law I=I_L+Iz so we get
    Iz_min=I_dash-I_L_dash

printf("maximum zener current = %.3f mA \n",Iz_max)
printf("minimum zener current = %.3f mA \n",Iz_min)
```

Scilab code Exa 9.22 Example 22

```
//chapter9
//example9.22
//page172

Ei=12 // V
Vz=7.2 // V
Eo=Vz
Iz_min=10d-3 // A
Il_max=100d-3 // A

// we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz
   occurs when Il is maximum so
R=(Ei-Eo)/(Iz_min+Il_max)

printf("required series resistance = %.3f ohm \n",R)

// on inserting this series resistance the output
   voltage will remain constant at 7.2 V

// the accurate answer is 43.636 ohm but in book it
   is given as 43.5 ohm
```

Scilab code Exa 9.23 Example 23

```
//chapter9
//example9.23
//page172

Ei=22 // V
Vz=18 // V
```

```
Rl=18 // ohm
Eo=Vz
Iz_min=200d-3 // A

// Zener current will be min when input voltage is
   min

// load current is
Il_max=Vz/Rl

// we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz
   occurs when Il is maximum so
R=(Ei-Eo)/(Iz_min+Il_max)

printf("required series resistance = %.3f ohm \n",R)

// on inserting this series resistance the output
   voltage will remain constant at 18 V
```

Scilab code Exa 9.24 Example 24

```
//chapter9
//example9.24
//page172

Ei=13 // V
Vz=10 // V
Eo=Vz
Iz_min=15d-3 // A
Il_max=85d-3 // A

// Zener current will be min when input voltage is
   min

// we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz
```

```
    occurs when Il is maximum so
R=(Ei-Eo)/(Iz_min+Il_max)

printf("required series resistance = %.3f ohm \n",R)
```

Scilab code Exa 9.25 Example 25

```
//chapter9
//example9.25
//page173

Ei=45 // V
Vz1=15 // V
Vz2=15 // V
Iz=200d-3 // current rating for each zener in ampere

Eo=Vz1+Vz2

R=(Ei-Eo)/Iz

printf("regulated output voltage = %.3f V \n",Eo)
printf("required series resistance = %.3f ohm \n",R)
```

Scilab code Exa 9.26 Example 26

```
//chapter9
//example9.26
//page173

Ei=45 // V
Vz1=10 // V
Vz2=10 // V
Vz3=10 // V
```

```
Iz=1000d-3 // current rating for each zener in
    ampere

Eo=Vz1+Vz2+Vz3

R=(Ei-Eo)/Iz

printf("required series resistance = %.3f ohm \n",R)

// since zener diode is not available in xcos,
    simple diodes are used to represent zener diode
    in the circuit made in xcos
```

Scilab code Exa 9.27 Example 27

```
//chapter9
//example9.27
//page174

R=200 // ohm
Rl=2000 // ohm
Eo=30 // V

// for minimum input voltage i.e. Iz=0
Il=Eo/Rl
I=Il // since Iz=0
Vin_min=Eo+I*R

// for maximum input voltage i.e. Iz=25 mA
Iz=25d-3 // A
Il_dash=Eo/Rl
I_dash=Il_dash+Iz
Vin_max=Eo+I_dash*R

printf("minimum input voltage = %.3f V \n",Vin_min)
```

```
printf("maximum input voltage = %.3f V \n",Vin_max)
printf("thus range of input = %.3f to %.3f V \n",Vin_min,Vin_max)
```

Scilab code Exa 9.28 Example 28

```
//chapter9
//example9.28
//page174

Ei=16 // V
Vz=12 // V    since we want to ragulate at 12 V
Eo=Vz
Iz_min=0 // A
Il_max=200d-3 // A

// Zener current will be min when input voltage is min

// we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz occurs when Il is maximum so
R=(Ei-Eo)/(Iz_min+Il_max)

Izm=Il_max
Pzm=Vz*Izm

printf("Zener voltage = %.3f V \n",Vz)
printf("required series resistance = %.3f ohm \n",R)
printf("maximum power rating of zener diode = %.3f W \n",Pzm)
```

Chapter 10

Special purpose diodes

Scilab code Exa 10.1 required series resistor

```
//chapter10
//example10.1
//page182

Vs=10 // V
Vd=1.6  // V
If=20d-3 // A

Rs=(Vs-Vd)/If

printf("required series resistor = %.3f ohm",Rs)
```

Scilab code Exa 10.2 current through LED

```
//chapter10
//example10.2
//page183

```

```
Vs=15 // V
Vd=2 // V
Rs=2.2d3 // ohm

If=(Vs-Vd)/Rs

printf("current through LED = %.3f A or %.3f mA",If,
    If*1000)
```

Scilab code Exa 10.3 dark resistance

```
//chapter10
//example10.3
//page187

//from graph, we see that for zero illumination, the
    reverse current i.e. dark current is 50 micro
    ampere

Ir=50d-6 // A
Vr=10 // V

Rr=Vr/Ir

printf("dark resistance = %.3f ohm or %.3f kilo ohm"
    ,Rr,Rr/1000)
```

Scilab code Exa 10.4 reverse current

```
//chapter10
//example10.4
//page188

```

```
m=37.4 // microA/mW/cm^2
E=2.5 // mW/cm^2

//since reverse current = sensitivity*illumination we can write
Ir=m*E

printf("reverse current = %.3f micro ampere",Ir)
```

Scilab code Exa 10.5 resonant frequency

```
//chapter10
//example10.5
//page192

L=1d-3 // H
C=100d-12 // F

fr=1/(2*%pi*(L*C)^0.5)

printf("resonant frequency = %.3f Hz or %.3f kHz",fr,fr/1000)
```

Scilab code Exa 10.11 base current

```
//chapter11
//example11.11
//page210

V_Rc=1
gain_beta=45
Rc=1 // kilo ohm

```

```
Ic=V_Rc/Rc
//since gain_beta=Ic/Ib
Ib=Ic/gain_beta

printf("base current = %.3f mA",Ib)
```

Chapter 11

Transistors

Scilab code Exa 11.1 voltage amplification

```
//chapter11
//example11.1
//page202

Rin=20 //ohm
Rout=100d3 //ohm
Rc=1d3 //ohm
signal=500d-3 //V

Ie=signal/Rin // A
Ic=Ie
Vout=Ic*Rc
Av=Vout/signal

printf("voltage amplification = %.2f \n",Av)
```

Scilab code Exa 11.2 base current

```
//chapter11
//example11.2
//page205

Ie=1 //mA
Ic=0.95 //mA

// since Ie=Ib+Ic we get
Ib=Ie-Ic

printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.3 base current

```
//chapter11
//example11.3
//page205

alpha=0.9
Ie=1 //mA

// since alpha=Ic/Ie we get

Ic=alpha*Ie

// since Ie=Ic+Ib we get

Ib=Ie-Ic

printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.4 amplification factor

```
//chapter11
//example11.4
//page205

Ic=0.95
Ib=0.05

Ie=Ib+Ic
alpha=Ic/Ie

printf("amplification factor = %.3f \n",alpha)
```

Scilab code Exa 11.5 collector current

```
//chapter11
//example11.5
//page205

Ie=1 //mA
alpha=0.92
Icbo=50d-3 //mA

Ic=alpha*Ie+Icbo

printf("collector current = %.3f mA \n",Ic)
```

Scilab code Exa 11.6 base current

```
//chapter11
//example11.6
//page205

alpha=0.95
```

```
V_Rc=2 // V
Rc=2 //kilo ohm

Ic=V_Rc/Rc // mA

// since alpha=Ic/Ie
Ie=Ic/alpha

// since Ie=Ib+Ic
Ib=Ie-Ic

printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.7 collector curent

```
//chapter11
//example11.7
//page206

Vbe=0.7 // V
Vcc=18 // V
Vee=8 // V
Rc=1.2 // kilo ohm
Re=1.5 //kilo ohm

// by Kirchoff's voltage law to emitter side loop,
   we get Vee=Ie*Re+Vbe so
Ie=(Vee-Vbe)/Re
Ic=Ie // nearly

// by Kirchoff's voltage law to collector side loop,
    we get Vcc=Ic*Rc=Vcb so
Vcb=Vcc-Ic*Rc

printf("collector curent  = %.3f mA \n",Ic)
```

```
printf("collector base voltage = %3f V \n",Vcb)
```

Scilab code Exa 11.8 Example 8

```
//chapter11
//example11.8
//page209

alpha1=0.9
alpha2=0.98
alpha3=0.99

beta1=alpha1/(1-alpha1)
beta2=alpha2/(1-alpha2)
beta3=alpha3/(1-alpha3)

printf("for alpha=0.9, beta=%.1f \n",beta1)
printf("for alpha=0.98, beta=%.1f \n",beta2)
printf("for alpha=0.99, beta=%.1f \n",beta3)
```

Scilab code Exa 11.9 Example 9

```
//chapter11
//example11.9
//page210

gain_beta=50
Ib=20d-3 // mA

// since gain_beta = Ic/Ib we get
Ic=gain_beta*Ib
Ie=Ic+Ib

```

```
printf("emitter current = %.3f mA \n",Ie)
```

Scilab code Exa 11.10 collector current

```
//chapter11
//example11.10
//page210

gain_beta=49
Ib=240d-3 // mA
Ie=12 // mA

alpha=gain_beta/(1+gain_beta)
Ic=alpha*Ie // or Ic=gain_beta*Ib

printf("collector current = %.3f mA \n",Ic)
```

Scilab code Exa 11.11 base current

```
//chapter11
//example11.11
//page210

V_Rc=1
gain_beta=45
Rc=1 // kilo ohm

Ic=V_Rc/Rc
//since gain_beta=Ic/Ib
Ib=Ic/gain_beta

printf("base current = %.3f mA",Ib)
```

Scilab code Exa 11.12 collector emitter voltage

```
//chapter11
//example11.12
//page210

Rc=800d-3 // kilo ohm
V_Rc=0.5 // V
Vcc=8 // V
alpha=0.96

Vce=Vcc-V_Rc
Ic=V_Rc/Rc // mA
gain_beta=alpha/(1-alpha)
Ib=Ic/gain_beta

printf("collector emitter voltage = %.3f V \n",Vce)
printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.13 base current

```
//chapter11
//example11.13
//page211

Ic=1000 // micro ampere
// when emitter circuit is open, leakage current =
   Icbo so
Icbo=0.2 // micro ampere

// when base is open, leakage current = Iceo so
Iceo=20 // micro ampere
```

```

//since Iceo=Icbo/(1-alpha) we get
alpha=1-(Icbo/Iceo)

// since Ic=alpha*Ie+Icbo we get
Ie=(Ic-Icbo)/alpha
Ib=Ie-Ic

printf("alpha = %.3f \n",alpha)
printf("emitter current = %.3f micro ampere \n",Ie)
printf("base current = %.3f micro ampere \n",Ib)
```

Scilab code Exa 11.14 dc load line

```
//chapter11
//example11.14
//page218

Vcc=12.5 // V
Rc=2.5 // kilo ohm

// we know that Vce=Vcc-Ic*Rc
// when Ic=0, Vce=Vcc i.e. 12.5V
// when Vce=0, Ic=Vcc/Rc i.e.5mA

// so equation of load line becomes Ic=-0.4*Vce+5
x=linspace(0,12.5,5)
y=-0.4*x+5
clf()
xtitle("dc load line","Vce(volts)","Ic(mA)")
plot2d(x,y,style=3,rect=[0,0,13,6])
```

Scilab code Exa 11.15 Q point

```
//chapter11
//example11.15
//page219

Vcc=12 // V
Rc=6 // kilo ohm

// we know that Vce=Vcc-Ic*Rc
// when Ic=0, Vce=Vcc i.e. 12V
// when Vce=0, Ic=Vcc/Rc i.e.2mA

// so equation of load line becomes Ic=-(1/6)*Vce+2
x=linspace(0,12,5)
y=-(1/6)*x+2
clf()
xtitle("dc load line","Vce(volts)","Ic(mA)")
plot2d(x,y,style=3,rect=[0,0,13,6])


// for Q point
Ib=20d-3 // mA
gain_beta=50

Ic=gain_beta*Ib
Vce=Vcc-Ic*Rc

printf("Q point = %.3f V and %.3f mA i.e. (%.3f,%.3f) \n",Vce,Ic,Vce,Ic)
```

Scilab code Exa 11.16 collector load

```
//chapter11
//example11.16
//page219

```

```
Vcc=10
Ic=1 // mA
Rc1=4 // kilo ohm
Rc2=5 // kilo ohm

Vce1=Vcc-Ic*Rc1
Vce2=Vcc-Ic*Rc2

printf("for collector load = 4 kilo ohm, operating
    point is %.3f V,%.3f mA \n",Vce1,Ic)
printf("for collector load = 5 kilo ohm, operating
    point is %.3f V,%.3f mA \n",Vce2,Ic)
```

Scilab code Exa 11.17 input resistance

```
//chapter11
//example11.17
//page222

del_Vbe=200 //mV
del_Ib=100 // micro ampere

Ri=del_Vbe/del_Ib

printf("input resistance = %.3f kilo ohm \n",Ri)
```

Scilab code Exa 11.18 output resistance

```
//chapter11
//example11.18
//page222

Vce2=10 // V
```

```
Vce1=2 // V
Ic1=2 // mA
Ic2=3 // mA

del_Vce=Vce2-Vce1 // V
del_Ic=Ic2-Ic1 // mA

Ro=del_Vce/del_Ic

printf("output resistance = %.3f kilo ohm \n",Ro)
```

Scilab code Exa 11.19 voltage gain

```
//chapter11
//example11.19
//page223

Rc=2 // kilo ohm
Ri=1 // kilo ohm
gain_beta=50

// for single stage , R_AC=Rc so voltage gain becomes
Av=gain_beta*Rc/Ri

printf("voltage gain = %.3f \n",Av)
```

Scilab code Exa 11.20 saturation collector current

```
//chapter11
//example11.20
//page224

Vcc=20 // V
```

```
Rc=1 // kilo ohm

// for saturation collector current, knee voltage becomes 0V so we get
Ic_sat=Vcc/Rc

// it can be seen from the circuit that cut-off voltage (i.e. when Ib=0) equals Vcc itself
Vce_cutoff=Vcc

// the equation of load line becomes Ic=-Vce+20

clf()
x=linspace(0,20,5)
y=-x+20
plot2d(x,y,style=3,rect=[0,0,21,21])
xtitle("dc load line","Vce(volts)","Ic(mA)")

printf("saturation collector current = %.3f mA \n",Ic_sat)
printf("cut-off collector emitter voltage = %.3f V \n",Vce_cutoff)
```

Scilab code Exa 11.21 maximum allowable collector current

```
//chapter11
//example11.21
//page225

Vce=20 // V
Pd=100 // mW

// since Pd=Vce*Ic we get
Ic=Pd/Vce

```

```
printf("maximum allowable collector current = %.3f
   mA \n ",Ic)
```

Chapter 12

Transistor biasing

Scilab code Exa 12.1 circuit diagram

```
// chapter 12
// example 12.1
// page 235

V_CC=6 // V
R_C=2.5 // kilo ohm

// for faithful amplification V_CE should not be
    less than V_CC for Si transistor so
V_max=V_CC-1
I_max=V_max/R_C

// As negative and positive half cyces of input are
    equal, change in collector current will be equal
    and opposite so
I_min=I_max/2

printf("Maximum allowable collector current = %.3f
   mA \n",I_max)
printf("Minimum zero signal collector current = %.3f
    mA \n",I_min)
```

```

// the circuit diagram is constructed on xcos and its screenshot has been taken.
// the waveform given can not be obtained in xcos unless we assume necessary values as data is insufficient for plotting graph in scilab.
// so waveform is constructed as below

clf()
x=linspace(1,5*%pi,100)
[t]=sin(x)+1
plot(x,[t])
xtitle("max and min allowable collector currents","t","i_c (mA)")
```

Scilab code Exa 12.2 maximum input signal voltage

```
//chapter12
//example12.2
//page236

Vcc=13 // V
V_knee=1 // V
Rc=4 // kilo ohm
gain_beta=100

V_Rc=Vcc-V_knee
Ic=V_Rc/Rc
Ib=Ic/gain_beta
Vs=Ic/5 // since Ic/Vs = 5 mA/V given

printf("maximum input signal voltage = %.3f V or %.3f mV \n",Vs,Vs*1000)
```

Scilab code Exa 12.3 Example 3

```
//chapter12
//example12.3
//page240

Vbb=2 // V
Vcc=9 // V
Rc=2 // kilo ohm
Rb=100 // kilo ohm
gain_beta=50

// by Kirchoff voltage law on base side, we get Ib*Rb+Vbe=Vbb so
Ib=Vbb/Rb // Vbe is negligible
Ic=gain_beta*Ib

// by Kirchoff voltage law on collector side, we get Ic*Rc+Vce=Vcc so
Vce=Vcc-Ic*Rc

// now for Rb=50 kilo ohm
Rb2=50 // kilo ohm

// since Rb is halved, Ib is doubled so
Ib2=2*Ib
Ic2=Ib2*gain_beta
Vce2=Vcc-Ic2*Rc

printf("for Rb = 100 kilo ohm, collector current = %.3f mA \nand collector emitter voltage = %.3f V \n \n",Ic,Vce)
printf("for Rb = 50 kilo ohm, collector current = %.3f mA \nand collector emitter voltage = %.3f V \
```

```
   n",Ic2,Vce2)
```

Scilab code Exa 12.4 Example 4

```
//chapter12
//example12.4
//page241

Vcc=6 // V
Rb=530 // kilo ohm
Rc=2 // kilo ohm
gain_beta=100
Vbe=0.7 // V

// when Ic=0, Vce=Vcc i.e. Vce=6 and when Vce=0, Ic=
   Vcc/Rc i.e. Ic=6/2
// so equation of load line becomes Ic=-0.5*Vce+3

x=linspace(0,6,5)
y=-0.5*x+3
plot2d(x,y,style=3,rect=[0,0,7,4])
xtitle("dc load line","Vce(volts)","Ic(mA)")

// since Vcc=Ib*Rb+Vbe we get
Ib=(Vcc-Vbe)/Rb
Ic=Ib*gain_beta
Vce=Vcc-Ic*Rc

printf("the operating point is %.3f V and %.3f mA \n
   ",Vce,Ic)

stability_factor=gain_beta+1

printf("stability factor=%.1f \n",stability_factor)
```

Scilab code Exa 12.5 Example 5

```
//chapter12
//example12.5
//page242

Vcc=12 // V
gain_beta=100
Vbe=0.3 // V
Ic=1 // mA

// since gain_beta=Ic/Ib
Ib=Ic/gain_beta

// since Vcc=Ib*Rb+Vbe we get
Rb=(Vcc-Vbe)/Ib

gain_beta2=50

// since Vcc=Ib*Rb+Vbe we get
Ib2=(Vcc-Vbe)/Rb
Ic2=Ib2*gain_beta2

printf("for beta = 100, base resistor = %.3f kilo
    ohm \n",Rb)
printf("for beta = 50, zero signal collector current
     for same Rb is = %.3f mA \n",Ic2)
```

Scilab code Exa 12.6 Example 6

```
//chapter12
//example12.6
```

```
//page242

Vcc=10 // V
R_B=1d3 // kilo ohm
R_E=1 // kilo ohm
Vbe=0 // since it is negligible
gain_beta=100

// by Kirchoff voltage law to base side we get Vcc=
    I_B*R_B+Vbe+I_E*R_E
// but I_E=I_B+I_C and I_C=gain_beta*I_B
// so we get Vcc=I_B*R_B+Vbe+R_E*I_B*(1+gain_beta)
// making I_B as subject we get

I_B=(Vcc-Vbe)/(R_B+R_E*(1+gain_beta)) // in ampere
I_C=gain_beta*I_B // in ampere
I_E=I_C+I_B // in ampere

printf("base current = %.4f mA \n",I_B)
printf("collector current = %.4f mA \n",I_C)
printf("emitter current = %.4f mA \n",I_E)
```

Scilab code Exa 12.7 Example 7

```
//chapter12
//example12.7
//page243

V_CC=15 // V
gain_beta=100
V_BE=0.6 // V
V_CE=8  // V
I_C=2 // mA

// here V_CC=V_CE+I_C*R_C so we get
```

```
R_C=(V_CC-V_CE)/I_C

I_B=I_C/gain_beta

// also V_CC=I_B*R_B+V_BE so we get
R_B=(V_CC-V_BE)/I_B

printf("collector resistance = %.3f kilo ohm \n",R_C
   )
printf("base resistance = %.3f kilo ohm \n",R_B)
```

Scilab code Exa 12.8 Example 8

```
//chapter12
//example12.8
//page245

V_CC=20 // V
R_B=100 // kilo ohm
R_C=1 // kilo ohm
V_BE=0.7 // V
gain_beta=100

// we know that R_B=(V_CC-V_BE-gain_beta*R_C*I_B)/
   I_B so we get
I_B=(V_CC-V_BE)/(R_B+gain_beta*R_C)

I_C=gain_beta*I_B

V_CE=V_CC-I_C*R_C

printf("operating point is %.3f V, %.3f mA \n",V_CE,
   I_C)

// the accurate answer is 10.35V,9.65mA but in book
```

it is given as 10.4V,9.6mA

Scilab code Exa 12.9 Example 9

```
//chapter12
//example12.9
//page245

V_CC=12 // V
gain_beta1=100
gain_beta2=50
V_BE=0.3 // V
V_CE=8  // V
I_C=1 // mA

// here V_CC=V_CE+I_C*R_C so we get
R_C=(V_CC-V_CE)/I_C

I_B=I_C/gain_beta1

// we know that R_B=(V_CC-V_BE-gain_beta1*R_C*I_B)/
   I_B so
R_B=(V_CC-V_BE-gain_beta1*R_C*I_B)/I_B


// for gain_beta=50 i.e. gain_beta2

// we know that R_B=(V_CC-V_BE-gain_beta2*R_C*I_B)/
   I_B so we get
I_B2=(V_CC-V_BE)/(R_B+gain_beta2*R_C)

I_C2=gain_beta2*I_B2

V_CE2=V_CC-I_C2*R_C

```

```
printf("for beta=100,required base resistance = %.3f
    kilo ohm \n",R_B)
printf("for beta=50,new operating point is %.3f V, %
   .3f mA \n",V_CE2,I_C2)
```

Scilab code Exa 12.10 base resistance

```
//chapter12
//example12.10
//page246

V_BE=0.7 // V
gain_beta=100
I_C=1 // mA
V_CE=2 // V

I_B=I_C/gain_beta

// since V_CE=V_BE+V_CB we get
V_CB=V_CE-V_BE

R_B=V_CB/I_B

printf("base resistance=%.3f kilo ohm \n",R_B)
```

Scilab code Exa 12.11 voltage across

```
//chapter12
//example12.11
//page248

Vcc=15 // V
Re=2 // kilo ohm
```

```
Rc=1 // kilo ohm
gain_beta=100
Vbe=0.7 // V
R1=10 // kilo ohm
R2=5 // kilo ohm

// when Ic=0, Vce=Vcc i.e. Vce=6 and when Vce=0, Ic=Vcc/(Rc+Re) i.e. Ic=15/(1+2)
// so equation of load line becomes Ic=-(1/3)*Vce+5

clf()
x=linspace(0,15,5)
y=-(1/3)*x+5
plot2d(x,y,style=3,rect=[0,0,16,6])
xtitle("dc load line","Vce(volts)","Ic(mA)")

V2=Vcc*R2/(R1+R2) // voltage across R2 i.e. 5 kilo ohm
Ie=(V2-Vbe)/Re
Ic=Ie
Vce=Vcc-Ic*(Rc+Re)

printf("the operating point is %.3f V and %.3f mA \n",Vce,Ic)
```

Scilab code Exa 12.12 thevenin

```
//chapter12
//example12.12
//page249

Vcc=15 // V
Re=2 // kilo ohm
Rc=1 // kilo ohm
gain_beta=100
```

```
Vbe=0.7 // V
R1=10 // kilo ohm
R2=5 // kilo ohm

Eo=Vcc*R2/(R1+R2)
Ro=R1*R2/(R1+R2)

printf("thevenin voltage = %.3f V \n",Eo)
printf("thevenin resistance = %.3f kilo ohm \n",Ro)

// here Eo=Ib*Ro+Vbe+Ie*Re
// now considering Ie=gain_beta*Ib, and making Ib as subject we get
// Ib=(Eo-Vbe)/(Ro+gain_beta*Re)
// Ic=gain_beta*Ib=gain_beta*(Eo-Vbe)/(Ro+gain_beta*Re)
// dividing numerator and denominator by gain_beta we get
// Ic=(Eo-Vbe)/(Re+Ro/gain_beta)
// Ro/gain_beta is negligible compared to Re so
Ic=(Eo-Vbe)/Re
Vce=Vcc-Ic*(Rc+Re)

printf("the operating point is %.3f V and %.3f mA \n",Vce,Ic)
```

Scilab code Exa 19.12 Example 12

```
//chapter19
//example19.12
//page424

It=8.93 // A
Ic=8 // A

```

```
// we know that (It/Ic)^2=1+m^2/2 so making m as
   subject we get
m=(2*((It/Ic)^2-1))^0.5

printf("modulation factor = %.3f or %.3f percent \n"
   ,m,m*100)
```

Scilab code Exa 12.13 collector current

```
//chapter12
//example12.13
//page250

R1=50 // kilo ohm
R2=10 // kilo ohm
Re=1 // kilo ohm
Vcc=12 // V
Vbe1=0.1 // V
Vbe2=0.3 // V

V2=Vcc*R2/(R1+R2) // voltage across R2

// for Vbe=0.1 V
Ic1=(V2-Vbe1)/Re

// for Vbe=0.3 V
Ic2=(V2-Vbe2)/Re

printf("for V_BE=0.1 V, collector current = %.3f mA
   \n",Ic1)
printf("for V_BE=0.3 V, collector current = %.3f mA
   \n \n",Ic2)

Vbe_change=100*(Vbe2-Vbe1)/Vbe1
Ic_change=-100*(Ic2-Ic1)/Ic1 // negative sign since
```

```
    Ic decreases
printf("comment : if V_BE changes by %.5f percent , \ncollector current changes by %.3f percent \n",
    Vbe_change,Ic_change)
printf("so collector current is independent of
    transistor parameter variations \n")

// the change in V_BE is 200 percent not 300 percent
    .It is mistake in textbook
```

Scilab code Exa 12.14 collector emitter

```
//chapter12
//example12.14
//page251

Vcc=20 // V
Re=5 // kilo ohm
Rc=1 // kilo ohm
Vbe=0 // considering it as negligible
R1=10 // kilo ohm
R2=10 // kilo ohm

V2=Vcc*R2/(R1+R2)

// since V2=Vbe+Ie*Re so
Ie=(V2-Vbe)/Re
Ic=Ie

Vce=Vcc-Ic*(Rc+Re)
Vc=Vcc-Ic*Rc

printf("emitter current = %.3f mA \n",Ie)
printf("collector emitter voltage = %.3f V \n",Vce)
printf("collector potential = %.3f V \n",Vc)
```

Scilab code Exa 12.15 Kirchoff voltage law

```
//chapter12
//example12.15
//page252

R_C=2.2 // kilo ohm
V_CC=9 // V
gain_beta=50
V_BE=0.3 // V
I_C=2 // mA
V_CE=3 // V

I_B=I_C/gain_beta
I1=10*I_B

// I1=V_CC/(R1+R2) so let Rt=R1+R2 thus we get
Rt=V_CC/I1

// by Kirchoff voltage law to collector side we get
// V_CC=I_C*R_C+V_CE+I_E*R_E and also we have I_C=
    I_E so
// V_CC=I_C*R_C+V_CE+I_C*R_E so making R_E as
    subject we get
R_E=((V_CC-V_CE)/I_C)-R_C // in kilo ohm

V2=V_BE+I_C*R_E // since V_E=I_C*R_E
R2=V2/I1
R1=Rt-R2

printf("emitter resistance = %.3f ohm \n",R_E*1000)
printf("R1 = %3f kilo ohm \n",R1)
printf("R2 = %3f kilo ohm \n",R2)
```

Scilab code Exa 12.16 collector resistance

```
//chapter12
//example12.16
//page252

alpha=0.985
V_BE=0.3 // V
V_CC=16 // V
V_CE=6 // V
I_C=2 // mA
R_E=2 // kilo ohm
R2=20 // kilo ohm

gain_beta=alpha/(1-alpha)
I_B=I_C/gain_beta

V_E=I_C*R_E
V2=V_BE+V_E
V1=V_CC-V2

I1=V2/R2
R1=V1/I1

V_RC=V_CC-V_CE-V_E
R_C=V_RC/I_C

printf("R1 = %.3f kilo ohm \n",R1)
printf("collector resistance = %.3f kilo ohm \n",R_C
   )
```

Scilab code Exa 12.17 thevenin

```
//chapter12
//example12.17
//page253

Vcc=15 // V
Re=2 // kilo ohm
Rc=1 // kilo ohm
gain_beta=100
Vbe=0.7 // V
R1=10 // kilo ohm
R2=5 // kilo ohm

Eo=Vcc*R2/(R1+R2)
Ro=R1*R2/(R1+R2)

printf("thevenin voltage = %.3f V \n",Eo)
printf("thevenin resistance = %.3f kilo ohm \n",Ro)

// here Eo=Ib*Ro+Vbe+Ie*Re
// now considering Ie=gain_beta*Ib, we can replace
    Ib=Ie/gain_beta
// Eo=(Ie/gain_beta)*Ro+Vbe+Ie*Re
// making Ie as subject we get
Ie=(Eo-Vbe)/(Re+Ro/gain_beta)

printf("emitter current = %.3f mA \n",Ie)
```

Scilab code Exa 12.18 kilo ohm

```
//chapter12
//example12.18
//page254

V_CC=10 // V
V_BE=0.2 // V
```

```
I_E=2 // mA
I_B=50d-3 // mA
R_E=1 // kilo ohm
R2=10 // kilo ohm

V2=V_BE+I_E*R_E
I2=V2/R2

I1=I2+I_B
V1=V_CC-V2
R1=V1/I1

printf("R1 = %.3f kilo ohm \n",R1)
```

Scilab code Exa 12.19 transistor

```
//chapter12
//example12.19
//page255

printf(" i) if R2 is shorted, base will be grounded. It will be \n    left without forward bias and transistor \n    will be cutoff so output is zero.\n \n")
printf(" ii) if R2 is open,forward bias will be very high. The \n    collector current will be very high and collector \n    emitter voltage will be very low. \n \n")
printf(" iii) if R1 is shorted, transistor will be in saturation \n    due to excessive forward bias. The base will be at \n    Vcc and emitter will be slightly below Vcc.\n \n")
printf(" iv) if R1 is open, transistor will be without forward bias.\n    Hence transistor will be cutoff i.e. output will be zero. \n")
```

Scilab code Exa 12.20 circuit is not mid-point biased

```
//chapter12
//example12.20
//page256

Vcc=8 // V
Rb=360 // kilo ohm
Rc=2 // kilo ohm
gain_beta=100
Vbe=0.7 // V

// when Ic=0, Vce=Vcc i.e. Vce=8 and when Vce=0, Ic=
    Vcc/Rc i.e. Ic=8/2
// so equation of load line becomes Ic=-0.5*Vce+4

clf()
x=linspace(0,8,5)
y=-0.5*x+4
plot2d(x,y,style=3,rect=[0,0,9,5])
xtitle("dc load line","Vce(volts)","Ic(mA)")

// since Vcc=Ib*Rb+Vbe we get
Ib=(Vcc-Vbe)/Rb
Ic=Ib*gain_beta
Vce=Vcc-Ic*Rc

printf("the operating point is %.3f V and %.3f mA \n
    ",Vce,Ic)
if Vce<Vcc/2+0.1 | Vce>Vcc/2-0.1 // check if V_CE is
     nearly half of V_CC
    printf("circuit is mid-point biased \n")
else
    printf("circuit is not mid-point biased. \n")
```

```
end
```

Scilab code Exa 12.21 circuit is not mid-point biased

```
//chapter12
//example12.21
//page257

V_CC=10 // V
R1=12 // kilo ohm
R2=2.7 // kilo ohm
V_BE=0.7 // V
R_E=180d-3 // kilo ohm
R_C=620d-3 // kilo ohm

V2=V_CC*R2/(R1+R2)
I_E=(V2-V_BE)/R_E
I_C=I_E
V_CE=V_CC-I_C*(R_C+R_E)

printf("the operating point is %.3f V and %.3f mA \n
   ",V_CE,I_C)
if V_CE<V_CC/2+0.1 | V_CE>V_CC/2-0.1  // check if
   V_CE is nearly half of V_CC
    printf("circuit is mid-point biased \n")
else
    printf("circuit is not mid-point biased. \n")
end

// the accurate answer for collector current is
   6.315 mA but in book it is given as 6.33 mA
```

Scilab code Exa 12.22 base current

```
//chapter12
//example12.22
//page257

V_CC=10 // V
R1=1.5 // kilo ohm
R2=0.68 // kilo ohm
R_E=0.24 // kilo ohm
V_BE=0.7 // V
beta_min=100
beta_max=400

V2=V_CC*R2/(R1+R2)
I_E=(V2-V_BE)/R_E
I_C=I_E

beta_avg=(beta_min*beta_max)^0.5
I_B=I_E/(beta_avg+1)

printf("base current = %f micro ampere \n",I_B*1000)

// the accurate answer for base current is 50.151
    micro ampere but in book it is given as 49.75
    micro ampere
```

Scilab code Exa 12.23 collector cutoff current

```
//chapter12
//example12.23
//page258

gain_beta=40
I_C1=2 // mA
t1=25 // degrees
t2=55 // degrees
```

```
I_CBO1=5d-3 // mA

// for I_CBO=5 micro ampere at 25 degrees
I_CEO1=(1+gain_beta)*I_CBO1

I_CBO2=I_CBO1*2^((t2-t1)/10) // since it doubles every 10 degrees. So for t2-t1, it becomes 2^((t2-t1)/10) times.
I_CEO2=(1+gain_beta)*I_CBO2
I_C2=I_CEO2+I_C1
I_C_change=100*(I_C2-I_C1)/I_C1

// for I_CBO=0.1 micro ampere at 25 degrees
t1_dash=25 // degrees
t2_dash=55 // degrees
I_CBO1_dash=0.1d-3 // mA
I_C1_dash=2 // mA

I_CBO2_dash=I_CBO1_dash*2^((t2-t1)/10) // since it doubles every 10 degrees. So for t2-t1, it becomes 2^((t2-t1)/10) times.
I_CEO2_dash=(1+gain_beta)*I_CBO2_dash
I_C2_dash=I_CEO2_dash+I_C1_dash
I_C_change_dash=100*(I_C2_dash-I_C1_dash)/I_C1_dash

printf("collector cutoff current = %.3f mA \n \n",I_CEO1)
printf("percent change in zero signal current given that \nI_CBO=5 micro ampere at 25 degree is = %.3f percent \n \n",I_C_change)
printf("percent change in zero signal current given that \nI_CBO=0.01 micro ampere at 25 degree is = %.3f percent \n",I_C_change_dash)
```

Scilab code Exa 12.24 base current

```
//chapter12
//example12.24
//page259

alpha=0.99
I_E=1 // mA
t1=27 // degrees
t2=57 // degrees
I_CBO1=0.02d-3 // mA

I_CBO2=I_CBO1*2^((t2-t1)/6) // since it doubles
    every 6 degrees. So for t2-t1, it becomes 2^((t2-
    t1)/6) times.

I_C=alpha*I_E+I_CBO2
I_B=I_E-I_C

printf("base current = %.1f micro ampere",I_B*1000)
```

Scilab code Exa 12.25 since base voltage is zero

```
//chapter12
//example12.25
//page261

printf("since base voltage is zero, it means that
    there is no path \nfor current in the base
    circuit. So the transistor will be off i.e. I_C
    =0,I_E=0. \nSo V_C=10V and V_E=0.\nSo obvious
    fault is R1 is open.\n")
```

Scilab code Exa 12.26 voltage at base

```
//chapter12
//example12.26
//page261

R1=18 // kilo ohm
R2=4.7 // kilo ohm
Re=1 // kilo ohm
Vcc=10 // V

V_B=Vcc*R2/(R1+R2)

printf("voltage at base = %.3f V \n",V_B)
printf("The fact that V_C=10V and V_E is nearly
    equal to V_B reveals \nthat I_C=0 and I_E=0.So
    I_B drops to zero.So obvious fault is R_E is open
    . \n")
```

Chapter 13

Single stage transistor amplifiers

Scilab code Exa 1.1 collector current

```
// chapter 13
// example 13.1
// page 272

Rc=4 // kilo ohm
Vcc=10 // V
Ib_zero=10d-3 // mA
Ib_max=15d-3 // mA
Ib_min=5d-3 // mA
gain_beta=100

Ic_zero=Ib_zero*gain_beta
Ic_max=Ib_max*gain_beta
Ic_min=Ib_min*gain_beta

Vc_zero=Vcc-Ic_zero*Rc
Vc_max=Vcc-Ic_max*Rc
Vc_min=Vcc-Ic_min*Rc

```

```
printf("As collector current increases from %.3f mA to %.3f mA \noutput voltage decreases from %.3f V to %.3f V \n",Ic_zero,Ic_max,Vc_zero,Vc_max)
printf("As collector current decreases from %.3f mA to %.3f mA \noutput voltage increases from %.3f V to %.3f V \n",Ic_max,Ic_min,Vc_max,Vc_min)
printf("Thus output voltage is 180 degrees out of phase from input voltage \n")

printf("Note : \ni) input voltage and input current are in phase \nii) input voltage and output current are in phase \niii) output voltage is 180 degrees out of phase with input voltage\n")


// plotting base current and collector current and output voltage in same graph using following code instead of xcos
clf()
x=linspace(0,2*%pi,100)
ib=5*sin(x)+10
ic=0.5*sin(x)+1
vc=-4*sin(x)+6
plot2d(x,ib,style=1,rect=[0,0,20,20])
xtitle("base current(micro ampere) - Black                         collector current(mA) - Blue                         output voltage(V) - Green                         ","t")
plot2d(x,ic,style=2,rect=[0,0,20,20])
plot2d(x,vc,style=3,rect=[0,0,20,20])
```

Scilab code Exa 1.2 Thevenin circuit

```
// chapter 13
// example 13.2
```

```
// page 274

printf("i)Refering to the Thevenin circuit, we see that voltage source \nis short and resistances except Rc and Re are bypassed.\nThus dc load = Rc + Re \n\n")
printf("  Refering to ac equivalent circuit, Rc is parallel with Rl.\nThus ac load = Rc*Rl/(Rc+Rl) \n \n \n")
printf("ii)Since Vcc=Vce+Ic*(Rc+Re) we get \n max Vce = Vcc and max Ic = Vcc/(Rc+Re) \n \n \n")
printf("iii)On applying ac signal, collector current and collector emitter \nvoltage change about Q point.\nMaximum collector current = Ic.\nMaximum positive swing of ac collector emitter voltage = Ic*R_AC \n So total maximum collector emitter voltage = Vce+Ic*R_AC \n\nMaximum positive swing of ac collector current = Vce/R_AC so \nTotal maximum collector current = Ic+Vce/R_AC \n")
```

Scilab code Exa 13.3 the operating point

```
// chapter 13
// example 13.3
// page 278

Vcc=15 // V
Re=2 // kilo ohm
Rc=1 // kilo ohm
Rl=1 // kilo ohm
Vbe=0.7 // V

// dc load line

    // when Ic=0, Vce=Vcc i.e. Vce=15 and when Vce
```

```
    =0, Ic=Vcc/(Rc+Re) i.e. Ic=15/3
// so equation of load line becomes Ic=-(1/3)*
    Vce+15

    clf()
x=linspace(0,15,5)
y=-(1/3)*x+5
plot2d(x,y,style=3,rect=[0,0,16,6])
xtitle("dc load line-green         ac load line-
    blue","collector emitter voltage(volts)","
    collector current(mA)")

V2=5 // V
// since voltage across R2 is V2=5 V and V2=Vbe+
    Ie*Re we get
Ie=(V2-Vbe)/Re
Ic=Ie
Vce=Vcc-Ic*(Rc+Re)

printf("the operating point is %.3f V and %.3f
    mA \n",Vce,Ic)


// ac load line

R_AC=Rc*Rl/(Rc+Rl) // ac load
V_ce=Vce+Ic*R_AC // maximum collector emitter
    voltage
I_c=Ic+Vce/R_AC // maximum collector current
// the equation of ac load line in terms of V_ce
    and I_c becomes
y=-(I_c/V_ce)*x+I_c
 plot2d(x,y,style=2,rect=[0,0,10,20])
```

Scilab code Exa 13.4 maximum collector

```
// chapter 13
// example 13.4
// page 279

Vcc=20 // V
Re=0 // kilo ohm, since given as negligible
Rc=10 // kilo ohm
Rl=30 // kilo ohm
Vbe=0.7 // V

Vce=10 // mV
Ic=1 // mA

// dc load line

    // when Ic=0, Vce=Vcc i.e. Vce=15 and when Vce=0, Ic=Vcc/(Rc+Re) i.e. Ic=20/10 mA
    // so equation of load line becomes Ic=-(1/10)*Vce+2

     clf()
    x=linspace(0,20,5)
    y=-(1/10)*x+2
    plot2d(x,y,style=3,rect=[0,0,21,6])
    xtitle("dc load line-green          ac load line-blue","collector emitter voltage(volts)","collector current(mA)")

// ac load line

    R_AC=Rc*Rl/(Rc+Rl) // ac load
    V_ce=Vce+Ic*R_AC // maximum collector emitter voltage
    I_c=Ic+Vce/R_AC // maximum collector current
    // the equation of ac load line in terms of V_ce and I_c becomes
    x=linspace(0,V_ce,10)
    y=-(I_c/V_ce)*x+I_c
```

```
    plot2d(x,y,style=2,rect=[0,0,21,6])
```

Scilab code Exa 13.5 plot

```
// chapter 13
// example 13.5
// page 280

printf("operating point is (8V,1mA). During positive half cycle of \nac signal collector current swings from 1 mA to 1.5 mA \nand collector emitter voltage swings from 8 V to 7 V.\nThis is at A.During negative half cycle of \nac signal collector current swings from 1 mA to 0.5 mA \nand collector emitter voltage swings from 8 V to 9 V.\nThis is at B. \n \n")

printf("Note : When ac signal is applied, ac signal collector current and \ncollector emitter voltage variations take place about Q point. \nAlso, operating point moves along load line.\n")

clf()
x=linspace(-3*%pi,-%pi,10)
plot(x,-0.5*sin(x)+1)

x=linspace(7,9,10)
plot(x,5-0.5*x)

x=linspace(-3*%pi,-%pi,10)
plot(-sin(x)+8,x)
plot(x,xgrid())
xtitle("collector current and collector emitter voltage swings","collector emitter voltage (volts)","collector current (mA)")
```

```
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";

// Some operations on entities created by plot ...
a=gca();
a.isoview='on';
a.children // list the children of the axes : here
   it is an Compound child composed of 2 entities
poly1= a.children.children(2); //store polyline
   handle into poly1
poly1.foreground = 4; // another way to change the
   style...
poly1.thickness = 3;  // ...and the tickness of a
   curve.
poly1.clip_state='off' // clipping control
a.isoview='off';
```

Scilab code Exa 13.6 voltage gain

```
//chapter13
//example13.6
//page282

Rc=2 // kilo ohm
Rl=0.5 // kilo ohm
Rin=1 // kilo ohm
gain_beta=60

R_AC=Rc*Rl/(Rc+Rl)
Av=gain_beta*R_AC/Rin

printf("voltage gain  = %.3f \n",Av)
```

Scilab code Exa 13.7 output voltage

```
//chapter13
//example13.7
//page282

Rc=10 // kilo ohm
Rl=10 // kilo ohm
Rin=2.5 // kilo ohm
gain_beta=100
Vin=1 // mV

R_AC=Rc*Rl/(Rc+Rl)
Av=gain_beta*R_AC/Rin

// since Av=Vout/Vin we get
Vout=Av*Vin

printf("output voltage = %.3f mV \n",Vout)
```

Scilab code Exa 13.8 input impedence

```
//chapter13
//example13.8
//page282

del_Ib=10d-3 // mA
del_Ic=1 // mA
del_Vbe=0.02 // V
Rc=5 // kilo ohm
Rl=10 // kilo ohm
gain_beta=60.0;
```

```
Ai=del_Ic/del_Ib
Rin=del_Vbe/del_Ib
R_AC=Rc*Rl/(Rc+Rl)
Av=gain_beta*R_AC/Rin
Ap=Av*Ai

printf("current gain = %.3f \n",Ai)
printf("input impedence = %.3f kilo ohm \n",Rin)
printf("ac load = %.3f kilo ohm \n",R_AC)
printf("voltage gain = %.3f \n",Av)
printf("power gain = %.3f \n",Ap)

// the accurate answer for voltage gain = 166.667
   and for power gain = 16666.667 but in book they
   are given as 165 and 16500 respectively.
```

Scilab code Exa 13.9 output voltage

```
//chapter13
//example13.9
//page283

Rc=3 // kilo ohm
Rl=6 // kilo ohm
Rin=0.5 // kilo ohm
Vin=1 // mV
gain_beta=50

R_AC=Rc*Rl/(Rc+Rl)
Av=gain_beta*R_AC/Rin

// since Av=Vout/Vin we get
Vout=Av*Vin

```

```
printf("output voltage = %.3f mV \n",Vout)
```

Scilab code Exa 13.10 circuit

```
//chapter13
//example13.10
//page283

R1=1 // kilo ohm
R2=2 // kilo ohm
Vt=6 // V

Vb=Vt*R1/(R1+R2)

if Vb==4
    printf("circuit is operating properly \n")
else
    printf("circuit is not operating properly
       because voltage at B should be %.1f V instead
        of 4 V \n",Vb)
end
```

Scilab code Exa 13.11 ac emitter resistance

```
//chapter13
//example13.11
//page284

R1=40 // kilo ohm
R2=10 // kilo ohm
Re=2 // kilo ohm
Vcc=10 // V
Vbe=0.7 // V
```

```
V2=Vcc*R2/(R1+R2) // voltage across R2
Ve=V2-Vbe // voltage across Re
Ie=Ve/Re
re_dash=25/Ie

printf("ac emitter resistance = %.3f ohm \n",re_dash
  )
```

Scilab code Exa 13.12 voltage gain

```
//chapter13
//example13.12
//page286

R1=150 // kilo ohm
R2=20 // kilo ohm
Re=2.2 // kilo ohm
Rc=12 // kilo ohm
Vcc=20 // V
Vbe=0.7 // V

V2=Vcc*R2/(R1+R2) // voltage across R2
Ve=V2-Vbe // voltage across Re
Ie=Ve/Re
re_dash=1d-3*25/Ie // in kilo ohm
Av=Rc/re_dash

printf("voltage gain = %.3f \n",Av)

// the accurate answer is 360.642
```

Scilab code Exa 13.13 voltage gain

```
//chapter13
//example13.13
//page287

Rc=12 // kilo ohm
Rl=6 // kilo ohm
re_dash=33.3d-3 // kilo ohm

R_AC=Rc*Rl/(Rc+Rl)
Av=R_AC/re_dash

printf("voltage gain = %.3f \n",Av)

// the accurate answer is 120.120
```

Scilab code Exa 13.14 input impedence of amplifier circuit

```
//chapter13
//example13.14
//page288

R1=45 // kilo ohm
R2=15 // kilo ohm
Re=7.5 // kilo ohm
Vcc=30 // V
Vbe=0.7 // V
gain_beta=200

V2=Vcc*R2/(R1+R2) // voltage across R2
Ve=V2-Vbe // voltage across Re
Ie=Ve/Re
re_dash=1d-3*25/Ie // in kilo ohm
Zin_base=gain_beta*re_dash
Zin=Zin_base*(R1*R2/(R1+R2))/(Zin_base+R1*R2/(R1+R2)
    )
```

```
printf("input impedence of amplifier circuit = %.3f
    kilo ohm \n",Zin)

// the accurate answer for input impedence is 3.701
    kilo ohm but in book it is given as 3.45 kilo ohm
```

Scilab code Exa 13.15 Class A

```
//chapter13
//example13.15
//page289

printf("i) Class A amplifier means that it raises
    voltage level of signal and its \nmode of
    operation is such that collector current flows
    for whole input signal. \n \n")
printf("ii) Audio voltage amplifier means it raises
    voltage level of audio signal \nand its mode of
    operation is class A. \n \n")
printf("iii) Class B power amplifier means that it
    raises power level of signal and its \nmode of
    operation is such that collector current flows
    for half cycle of input signal only. \n \n")
printf("iv) Class A transformer coupled power
    amplifier means that power amplification \nis
    being done, coupling is by tranformer and mode of
     operation is class A. \n")
```

Scilab code Exa 13.16 required input signal voltag

```
//chapter13
//example13.16
```

```
//page290

Ao=1000
Rout=1 // ohm
Rl=4 // ohm
Rin=2d3 // ohm
I2=0.5 // A

// here I2/I1=Ao*Rin/(Rout+Rl) so
I1=I2*(Rout+Rl)/(Ao*Rin)
V1=I1*Rin // in V

printf("required input signal voltage = %.3f mV \n",
   V1*1d3)
```

Scilab code Exa 13.17 magnitude of output voltage

```
//chapter13
//example13.17
//page291

Ao=1000
Es=10d-3 // V
Rs=3d3 // ohm
Rin=7d3 // ohm
Rout=15 // ohm
Rl=35 // ohm

I1=Es/(Rs+Rin)
V1=I1*Rin
Av=Ao*Rl/(Rout+Rl)
// since V2/V1=Av, we get
V2=V1*Av

P2=V2^2/Rl
```

```
P1=V1^2/Rin
Ap=P2/P1

printf("magnitude of output voltage = %.2f V \n",V2)
printf("power gain = %.2f \n",Ap)
```

Scilab code Exa 13.18 required signal voltage

```
//chapter13
//example13.18
//page292

Av=80
Ai=120
V2=1 // V
Rout=1 // ohm
Rl=2 // ohm

V1=V2/Av // in V

// Av=Ao*Rl/(Rout+Rl) and Ai=Ao*Rin/(Rout+Rl) so
// Av/Ai=Rl/Rin hence
Rin=Rl*Ai/Av

I1=V1/Rin // in mA
Ap=Av*Ai

printf("required signal voltage = %.2f mV and
    current = %.2f micro ampere \n",V1*1d3,I1*1d3)
printf("power gain = %.3f \n",Ap)
```

Chapter 14

Multistage transistor amplifiers

Scilab code Exa 14.1 gain

```
//chapter 14
//example 14.1
//page 301

Av=20*log10(30)

Pv=10*log10(100)

printf("voltage gain = %.3f db \n",Av)
printf("power gain = %.3f db \n",Pv)
```

Scilab code Exa 14.2 gain

```
//chapter 14
//example 14.2
//page 301

Ap1=40 // db
```

```
Ap2=43 // db

// since Ap = 10*log10(power_gain), we get
power_gain1=10^(Ap1/10)
power_gain2=10^(Ap2/10)

printf("power gain of 40 db = %.3f \n",power_gain1)
printf("power gain of 43 db = %.3f \n",power_gain2)

// the accurate answer for power gain of 43 db is
   19952 but in book it is given as 20000 db
```

Scilab code Exa 14.3 total voltage gain

```
//chapter 14
//example 14.3
//page 301

Av1=20*log10(100) // db
Av2=20*log10(200) // db
Av3=20*log10(400) // db

Av_total=Av1+Av2+Av3

printf("total voltage gain = %.3f db \n",Av_total)
```

Scilab code Exa 14.4 Example 4

```
//chapter 14
//example 14.4
//page 302

gain_abs=30
```

```
n=5

Ap1=10*log10(gain_abs) // db
Ap_tot=Ap1*n
Ap_f=Ap_tot-10 // db

printf("total power gain = %.3f db \n",Ap_tot)
printf("power gain with negative feedback = %.3f db
    \n",Ap_f)
```

Scilab code Exa 14.5 Example 5

```
//chapter 14
//example 14.5
//page 302

P1=1.5 // W
P2=0.3 // W
Pi=10d-3 // W

// power gain at 2 kHz
Ap1=10*log10(P1/Pi)

// power gain at 20 Hz
Ap2=10*log10(P2/Pi)

Ap_diff=Ap1-Ap2
printf("fall in gain from 2 kHz to 20 Hz = %.3f db \
    n",Ap_diff)
```

Scilab code Exa 14.6 output voltage

```
//chapter 14
```

```
//example 14.6
//page 302

Av=15 // db
V1=0.8 // V

// since db voltage gain Av=20*log10(V2/V1) making
   V2 as subject we get

V2=V1*10^(Av/20)

printf("output voltage = %.2f V \n",V2)
```

Scilab code Exa 14.7 minimum value of load resistance

```
//chapter 14
//example 14.7
//page 302

Ao_db=70 // db
Av_db=67 // db
Rout=1.5 // kilo ohm

// since 20*log(Ao)-20*log(Av)=Ao_db-Av_db we get
// 20*log(Ao/Av) = Ao_db-Av_db so
// Ao/Av = 10^((Ao_db-Av_db)/20)
// and also Ao/Av=1+Rout/Rl since Av/Ao=Rl/(Rl+Rout)

// so making Rl as subject we get
Rl=Rout/(10^((Ao_db-Av_db)/20)-1)

printf("minimum value of load resistance = %.3f kilo
    ohm \n",Rl)

// the accurate answer is 3.636 kilo ohm
```

Scilab code Exa 14.8 output voltage

```
//chapter 14
//example 14.8
//page 303

gain_db=40 // db
Vin=10d-3 // mV
Rl=1 // kilo ohm

// we know that Vout/Vin=10^(gain_db/20) so making
   Vout as subject we get
Vout=Vin*10^(gain_db/20)
P_load=Vout^2/Rl

printf("output voltage = %.3f V \n",Vout)
printf("load power = %.3f mW \n",P_load)
```

Scilab code Exa 14.9 frequency

```
// chapter14
// example14.9
// page 303

// figure given in book is for reference only. It is
    not required to solve the example since the
   required details are very clearly specified in
   the problem statement.
// moreover more data is needed to plot the graph
   given in book.

```

```
Av_max1=2000 // for 2 kHz
Av_sqrt_2=1414 // for 10 kHz and 50 Hz

percent_Av_max1=70.7*Av_max1/100
printf("70.7 percent of maximum gain 2000 is = %.3f \n",percent_Av_max1)

if Av_sqrt_2==percent_Av_max1
printf("we observe that 70.7 percent of max gain 2000 is 1414 \n")
printf("this gain 1414 is at 50 Hz and 10 kHz \n")
printf("so bandwidth = 50 Hz to 10 kHz \n \n")

printf("since frequency on lower side at which gain falls to \n70.7 percent is 50 Hz.So lower cutoff frequency = 50 Hz \n \n")
printf("since frequency on upper side at which gain falls to \n70.7 percent is 10 kHz.So upper cutoff frequency = 10 kHz \n \n")
else printf("data is insuficient for finding bandwidth and cutoff frequencies \n")
end
```

Scilab code Exa 14.10 gain

```
// chapter14
// example14.10
// page 305

Rc=500 // ohm
Rin=1d3 // ohm

// gain of second stage is 60 since it has no loading effect of any stage so
Av2=60
```

```
load1=Rc*Rin/(Rc+Rin)
Av1=Av2*load1/Rc
Av=Av1*Av2

printf("total gain = %.3f \n",Av)
printf("comment : gain of one stage=60.So total gain
    should be 60*60=%d but here it is %.3f.\nThis is
    because of loading effect of input impedence of
   second stage on first stage. \n",60*60,Av)
printf("So gain of first stage decreases.\nHowever,
   second stage has no loading effect of any next
   stage.So its gain does not decrease. \n")

// the accurate answer for total gain is 2400 but in
    book it is given as 2397
```

Scilab code Exa 14.11 gain

```
// chapter14
// example14.11
// page 306

Rin=1 // kilo ohm
Rc= 2 // kilo ohm
gain_beta=100

// since first stage has loading effect of input
   impedence of second stage, we get effective load
   of first stage as
R_AC=Rc*Rin/(Rc+Rin)
Av1=gain_beta*R_AC/Rin

// second stage has no loading effect so its gain
Av2=gain_beta*Rc/Rin
Av=Av1*Av2
```

```
printf("voltage gain of first stage = %.3f \n",Av1)
printf("voltage gain of second stage = %.3f \n",Av2)
printf("total voltage gain = %.3f \n",Av)

// the accurate answer for gain of first stage is
    66.667 and total gain is 13333.33 but in book
    they are given as 66 and 13200 respectively
```

Scilab code Exa 14.12 voltage gain

```
// chapter14
// example14.12
// page 307

Rin=1d3 // ohm
Rc= 10d3 // ohm
Rl=100 // ohm
gain_beta=100

// effective collector load is
R_AC=Rc*Rl/(Rc+Rl)
Av=gain_beta*R_AC/Rin

printf("voltage gain = %.3f \n",Av)
printf("comment : load is only 100 ohm so efective
    load of amplifier is too much reduced.\nThus
    voltage gain is very small.\n")
printf("In such cases we can use a step down
    transformer to serve the purpose. \n")
```

Scilab code Exa 14.13 biasing potential

```
// chapter14
// example14.13
// page 307

Vcc=20 // V
R3=10 // kilo ohm
R4=2.2 // kilo ohm
Rc=3.6 // kilo ohm

V_B=Vcc*R4/(R3+R4)

// replacing Cc by wire
Req=R3*Rc/(R3+Rc)
V_B2=Vcc*R4/(Req+R4)

printf("biasing potential before replacing Cc = %.3f V \n",V_B)
printf("biasing potential after replacing Cc = %.3f V \n \n",V_B2)
printf("thus biasing potential of second stage changes.\nThis could cause the transistor to saturate and it would not work as amplifier.\n")
printf("Also, we see the use of coupling capacitor to maintain \nindependent biasing potential for each stage.\nThis allows ac output from one stage to pass to next stage.\n")
```

Scilab code Exa 14.14 voltage gain

```
// chapter14
// example14.14
// page 308

Vcc=15 // V
R1=22 // kilo ohm
```

```
R2=3.3 // kilo ohm
R3=5 // kilo ohm
R4=1 // kilo ohm
R5=15 // kilo ohm
R6=2.5 // kilo ohm
R8=1 // kilo ohm
R3=5 // kilo ohm
R7=5 // kilo ohm
Rl=10 // kilo ohm
gain_beta=200
Vbe=0.7 // V

// for second stage
V_R6=Vcc*R6/(R6+R5)
V_R8=V_R6-Vbe
I_E2=V_R8/R8 // emitter current in R8
re_dash2=25d-3/I_E2
Zin_base=gain_beta*re_dash2
Zin=R5*(R6*Zin_base/(R6+Zin_base))/(R5+(R6*Zin_base
    /(R6+Zin_base)))
R_AC2=R7*Rl/(R7+Rl)
Av2=R_AC2/re_dash2

// for first stage
V_R2=Vcc*R2/(R2+R1)
V_R4=V_R2-Vbe
I_E1=V_R4/R4 // emitter current in R4
re_dash1=25d-3/I_E1
R_AC1=R3*Zin/(R3+Zin)
Av1=R_AC1/re_dash1

Av=Av1*Av2

printf("voltage gain of first stage = %.3f \n",Av1)
printf("voltage gain of second stage = %.3f \n",Av2)
printf("overall voltage gain= %.3f \n",Av)

// the accurate answers are voltage gain of first
```

```
   stage = 52.616, voltage gain of second stage =
   192.381, overall voltage gain= 10122.329. In book
    the answers are 53,191.4 and 10144
// respectively
```

Scilab code Exa 14.15 transformer

```
// chapter14
// example14.15
// page 311

// for maximum power transfer, primary impedence =
   transistor output impedence and secondary
   impedence = load impedence
Rp=1d3 // ohm
Rs=10 // ohm

// since Rp=(Np/Ns)^2*Rs, making Np/Ns i.e. n as
   subject we get
n=(Rp/Rs)^(0.5)

printf("required turn ratio = %d \n",n)

if n>1
    printf("transformer required is step down
       tranformer \n")
elseif n<1
    printf("transformer required is step up
       tranformer \n")
else // n=1
    printf("transformer is not required \n")
end
```

Scilab code Exa 14.16 voltage across external load

```
// chapter14
// example14.16
// page 312

Vp=10 // V
// for maximum power transfer , primary impedence =
   output impedence of aource
Rp=10d3 // ohm
Rs=16 // ohm

// since Rp=(Np/Ns)^2*Rs, making Np/Ns i.e. n as
   subject we get
n=(Rp/Rs)^(0.5)

// since Vs/Vp=Ns/Np, making Vs as subject we get
Vs=(1/n)*Vp
printf("required turn ratio = %d \n",n)
printf("voltage across external load = %.3f V \n",Vs
   )
```

Scilab code Exa 14.17 turn ratio for maximum power transfer

```
// chapter14
// example14.17
// page 312

Rp=300 //ohm
Rs=3 // ohm
Ro=3d3 // ohm

// since output resistance of transistor Ro=Rp+n^2*
   Rs for maximum power transfer , making n as
   subject we get
```

```
n=((Ro-Rp)/Rs)^0.5

printf("turn ratio for maximum power transfer = %d \
   n",n)
```

Scilab code Exa 14.18 inductance

```
// chapter14
// example14.18
// page 313

f=200 // Hz
Ro=10d3 // ohm, transistor output impedence
Zi2=2.5d3 // ohm, input impedence of next stage

// since Ro=2*%pi*f*Lp, making Lp as subject we get
Lp=Ro/(2*%pi*f)

// since Zi2=2*%pi*f*Ls, making Ls as subject we get
Ls=Zi2/(2*%pi*f)

printf("primary inductance = %.1f H \n",Lp)
printf("secondary inductance = %.1f H \n",Ls)
```

Scilab code Exa 14.19 inductance

```
// chapter14
// example14.19
// page 313

L=10d-6 // H
N=1 // turn
Lp=8 // H
```

```
Ls=2 // H

// since L is proportional to N^2, L=K*N^2 so making
    K as subject we get
K=L/N^2

// Lp=K*Np^2 so
Np=(Lp/K)^0.5

// Ls=K*Ns^2 so
Ns=(Ls/K)^0.5

printf("primary turns = %d \n",Np)
printf("secondary turns = %d \n",Ns)
```

Chapter 15

Transistor audio power amplifiers

Scilab code Exa 15.1 maximum collector current

```
//chapter15
//example15.1
//page321

V=12 // V
P=2 // W

// since P=V*Ic we get
Ic_max=P/V // in ampere

printf("maximum collector current = %.3f mA \n",
   Ic_max*1000)
```

Scilab code Exa 15.2 maximum collector current

```
//chapter15
```

```
//example15.2
//page321

V=12 // V
R=4 // kilo ohm

//since maximum collector current will flow when
   whole battery voltgage is dropped across Rc, we
   get
Ic_max=V/R

printf("maximum collector current = %.3f mA \n",
   Ic_max)
```

Scilab code Exa 15.3 ac output

```
//chapter15
//example15.3
//page321

P=50 // W
R=8 // ohm

// since p=V^2/R we get
V=(P*R)^0.5
I=V/R

printf("ac output voltage = %.3f V \n",V)
printf("ac output current = %.3f A \n",I)
```

Scilab code Exa 15.4 collector efficiency

```
//chapter15
```

```
//example15.4
//page325

Vcc=20 // V
Vbe=0.7 // V
Rb=1d3 // ohm
Rc=20 // ohm
gain=25

Ib=(Vcc-Vbe)/Rb
Ic=Ib*gain
Vce=Vcc-Ic*Rc

ib_peak=10d-3
ic_peak=gain*ib_peak
Po_ac=ic_peak^2*Rc/2
P_dc=Vcc*Ic
eta=(Po_ac/P_dc)*100

printf("operating point = %.3f V and %.3f mA \n",Vce
    ,Ic*1000)
printf("output power = %.3f W \n",Po_ac)
printf("input power = %.3f W \n",P_dc)
printf("collector efficiency = %.3f percent \n",eta)
```

Scilab code Exa 15.5 collector efficiency

```
//chapter15
//example15.5
//page328

Pdc=10 // W
Po=4 // W

eta=(Po/Pdc)*100
```

```
// maximum power dissipation in a transistor occurs
   under zero signal conditions so
P=Pdc

printf("collector efficiency = %.3f percent \n",eta)
printf("power rating of transistor = %.3f W \n",P)
```

Scilab code Exa 15.6 maximum ac power output

```
//chapter15
//example15.6
//page328

Rl=100 // ohm
n=10
Ic=100d-3 // ampere

Rl_1=n^2*Rl
Pmax=0.5*Ic^2*Rl_1

printf("maximum ac power output = %.3f W \n",Pmax)
```

Scilab code Exa 15.7 power dissipation

```
//chapter15
//example15.7
//page329

Vcc=5 // V
Ic=50d-3 // ampere

Pac_max=Vcc*Ic/2
```

```
Pdc=Vcc*Ic
Pdis=Pac_max*2
eta=(Pac_max/Pdc)*100

printf("maximum power output= %.3f mW \n",Pac_max
    *1000)
printf("power dissipation = %.3f mW \n",Pdis*1000)
printf("maximum collector efficiency = %.3f percent
    \n",eta)
```

Scilab code Exa 15.8 loudspeaker

```
//chapter15
//example15.8
//page329

del_Ic=100d-3 // ampere
del_Vce=12 // V
Rl=5 // ohm

//case 1 : louspeaker directly connected
V=del_Ic*Rl
P=V*del_Ic

//case2 : loudspreaker transformer coupled
R_primary=del_Vce/del_Ic // for maximum power
    transfer the primary resistance should be equal
    to R
n=(R_primary/Rl)^0.5
V_secondary=del_Vce/n
Il=V_secondary/Rl
P_1=Il^2*Rl

printf("case1 : loudspeaker connected directly \n
        power transferred to loudspeaker = %.3f mW \
```

```
   n",P*1000)
printf("case2 : loudspeaker is transformer coupled \
   n       power transferred to loudspeaker = %.3f mW
    \n",P_1*1000)
```

Scilab code Exa 15.9 current

```
//chapter15
//example15.9
//page331

Vcc=17.5 // V
ic_max=35d-3 // ampere
ic_min=1d-3 // ampere
IC=18 // ampere
gain=100
vce_max=30 // V
vce_min=5 // V
Rl=81.6 // ohm

IC=ic_min+((ic_max-ic_min)/2)
IB=IC/gain
VCE=vce_min+((vce_max-vce_min)/2)

Pdc=Vcc*IC
Vce=(vce_max-vce_min)/(2*2^0.5)
Ic=(ic_max-ic_min)/(2*2^0.5)
Pac=Vce*Ic

eta=(Pac/Pdc)*100

slope=(ic_max-ic_min)/(vce_min-vce_max)
Rl_dash=-1/slope
n=(Rl_dash/Rl)^0.5

```

```
printf("zero signal collector current = %.3f mA \n",IC*1000)
printf("zero signal base current = %.3f mA \n",IB*1000)
printf("dc power = %.3f mW and ac power = %.3f mW \n",Pdc*1000,Pac*1000)
printf("collector efficiency = %.3f percent \n",eta)
printf("transformer turn ratio = %.1f \n",n)
```

Scilab code Exa 15.10 maximum ambient temperature at which transistor can be operated

```
//chapter15
//example15.10
//page333

P_total=4 // W
T_Jmax=90 // degree celcius
theta=10 // degree celcius per watt

// P_total=(T_Jmax-Tamb)/theta so making Tamb as subject we get
Tamb=T_Jmax-P_total*theta

printf("maximum ambient temperature at which transistor can be operated = %.3f degree C \n",Tamb)
```

Scilab code Exa 15.11 maximum power dissipation

```
//chapter15
//example15.11
//page333
```

```

T_Jmax=90 // degree celcius
T_amb=30 // degree celcius

//case 1 : without heat sink
theta1=300 // degree celcius per watt
P_total1=(T_Jmax-T_amb)/theta1

//case 2 : with heat sink
theta2=60 // degree celcius per watt
P_total2=(T_Jmax-T_amb)/theta2

printf("case 1 : without heat sink \n   maximum
    power dissipation = %.3f mW \n",P_total1*1000)
printf("case 2 : with heat sink \n   maximum power
    dissipation = %.3f mW \n",P_total2*1000)
```

Scilab code Exa 15.12 allowed collector current

```
//chapter15
//example15.12
//page334

T_Jmax=200 // degree celcius
T_amb1=25 // degree celcius
T_amb2=75 // degree celcius
theta=20 // degree celcius per watt
Vcc=4 // V

P_total1=(T_Jmax-T_amb1)/theta
Ic1=P_total1/Vcc

P_total2=(T_Jmax-T_amb2)/theta
Ic2=P_total2/Vcc

```

```
printf("for ambient = 25 degree C, allowed collector
    current = %.3f A \n",Ic1)
printf("for ambient = 75 degree C, allowed collector
    current = %.3f A \n",Ic2)
```

Chapter 16

Amplifiers with negative feedback

Scilab code Exa 16.1 voltge gain with negative feedback

```
//chapter16
//example16.1
//page345

Av=3000
mv=0.01

Avf=Av/(1+Av*mv)
printf("voltge gain with negative feedback = %.3f \n
   ",Avf)

// accurate answer is 96.774 but in book it is given
   as 97
```

Scilab code Exa 12.2 fraction of output fed back to input

```
//chapter16
//example16.2
//page346

Av=140
Avf=17.5

// since Avf=Av/(1+Av*mv), making mv as subject we
   get
mv=(Av/Avf-1)/Av
printf("fraction of output fed back to input = %.3f
   \n",mv)
```

Scilab code Exa 16.3 overall gain

```
//chapter16
//example16.3
//page346

Av1=100
Avf1=50
Avf2=75

// since Avf=Av/(1+Av*mv), we get
mv=(Av1/Avf1-1)/Av1
Av2=Avf2/(1-mv*Avf2)

printf("fraction of output fed back to input = %.3f
   \n",mv)
printf("for overall gain = 75 and same fraction,
   required gain = %.3f \n",Av2)
```

Scilab code Exa 16.4 fraction of output fed back to input

```
//chapter16
//example16.4
//page346

Vo=10
Vi=0.25
Vif=0.5

Av=Vo/Vi
Avf=Vo/Vif

// since Avf=Av/(1+Av*mv), we get
mv=(Av/Avf-1)/Av

printf("fraction of output fed back to input = %.3f
    \n",mv)
```

Scilab code Exa 16.5 percentage reduction in gain

```
//chapter16
//example16.5
//page347

Av=50
Avf=25

// since Avf=Av/(1+Av*mv), we get
mv=(Av/Avf-1)/Av

// without feedback, gain falls from 50 to 40
Av1=50
Av2=40
reduction1=100*(Av1-Av2)/Av1

// with feedback
```

```
Av3=25
Av4=Av2/(1+mv*Av2)
reduction2=100*(Av3-Av4)/Av3

printf("percentage reduction in gain : \n with
    feedback = %.3f percent \n ",reduction1)
printf("without feedback = %.3f perent",reduction2)
```

Scilab code Exa 16.6 percentage change in gain

```
//chapter16
//example16.6
//page347

Av=100
mv=0.1

Avf=Av/(1+Av*mv)
mv=(Av/Avf-1)/Av

// fall in gain is 6dB so 20log(Av/Av1)=6
// making Av1 as subject we get
Av1=Av/exp(6*log(10)/20)
Avf_new=Av1/(1+Av1*mv)
change=100*(Avf-Avf_new)/Avf

printf("percentage change in gain = %.3f percent \n"
    ,change)

// the accurate answer is 8.297 percent but in book
    it is given as 8.36 percent
```

Scilab code Exa 16.7 voltage gain with negative feedback

```
//chapter16
//example16.7
//page348

A0=1000
Rout=100 // ohm
Rl=900
mv=1/50

Av=A0*Rl/(Rout+Rl)
Avf=Av/(1+Av*mv)
printf("voltage gain with negative feedback = %.3f \
   n",Avf)

// the accurate answer is 47.368 but in book it is
   given as 47.4
```

Scilab code Exa 16.8 voltage gain with negative feedback

```
//chapter16
//example16.8
//page351

Av=10000
R1=2 // kilo ohm
R2=18 // kilo ohm
Vi=1 // mV

mv=R1/(R1+R2)
Avf=Av/(1+Av*mv)
Vout=Avf*Vi

printf("feedback fraction  = %.1f \n",mv)

printf("voltage gain with negative feedback = %.1f \
```

```
n",Avf)

printf("output voltage = %.1f mV \n",Vout)
```

Scilab code Exa 16.9 Example 9

```
//chapter16
//example16.9
//page351

Av=10000
R1=10 // kilo ohm
R2=90 // kilo ohm
Zin=10 // kilo ohm
Zout=100d-3 // kilo ohm

mv=R1/(R1+R2)
Avf=Av/(1+Av*mv)
Zin_dash=(1+Av*mv)*Zin
Zout_dash=Zout/(1+Av*mv)

printf("feedbackfraction = %.1f \n",mv)

printf("voltage gain with negative feedback = %.1f \n",Avf)

printf("input impedence with feedback = %.3f kilo ohm or %.3f mega ohm \n",Zin_dash,Zin_dash/1000)

printf("output impedence with feedback = %f kilo ohm or %.3f ohm \n",Zout_dash,Zout_dash*1000)
```

Scilab code Exa 16.10 distortion of amplifier with negative feedback

```
//chapter16
//example16.10
//page352

Av=150
D=5/100
mv=10/100

Dvf=100*D/(1+Av*mv) // in percent

printf("distortion of amplifier with negative
    feedback = %.3f percent",Dvf)
```

Scilab code Exa 16.11 cutoff frequency with negative feedback

```
//chapter16
//example16.11
//page352

Av=1000
mv=0.01
f1=1.5 // kHz
f2=501.5 // kHz

f_1f=f1/(1+Av*mv)
f_2f=f2*(1+mv*Av)

printf("new lower cutoff frequency with negative
    feedback = %.3f kHz or %.3f Hz \n",f_1f,f_1f
    *1000)
printf("new upper cutoff frequency with negative
    feedback = %.3f kHz or %.3f MHz \n",f_2f,f_2f
    /1000)

// the accurate answers are 136.364 Hz and 5.516 MHz
```

```
but in book they are given as 136.4 Hz and 5.52
MHz respectively
```

Scilab code Exa 16.12 effective current gain of amplifier

```
//chapter16
//example16.12
//page353

Ai=200
mi=0.012

Aif=Ai/(1+mi*Ai)

printf("effective current gain of amplifier = %.3f \n",Aif)
```

Scilab code Exa 16.13 input impedence with negative feedback

```
//chapter16
//example16.13
//page354

Zin=15// kilo ohm
Ai=240
mi=0.015

Zin_dash=Zin/(1+mi*Ai)

printf("input impedence with negative feedback = %.3f kilo ohm \n",Zin_dash)
```

Scilab code Exa 16.14 output impedence with negative feedback

```
//chapter16
//example16.14
//page355

Zout=3 // kilo ohm
Ai=200
mi=0.01

Zout_dash=Zout*(1+mi*Ai)

printf("output impedence with negative feedback = %
    .3f kilo ohm \n",Zout_dash)
```

Scilab code Exa 16.15 Bandwidth with negative feedback

```
//chapter16
//example16.15
//page355

BW=400 // kHz
Ai=250
mi=0.01

BW_dash=BW*(1+mi*Ai)

printf("Bandwidth with negative feedback = %.3f kHz
    \n",BW_dash)
```

Scilab code Exa 16.16 voltage across

```
//chapter16
//example16.16
//page356

Vcc=18 // V
R1=16 // kilo ohm
R2=22 // kilo ohm
Vbe=0.7 // V
Re=910d-3 // kilo ohm

V2=Vcc*R2/(R1+R2)
Ve=V2-Vbe
Ie=Ve/Re

printf("voltage across Re = %.3f V \n",Ve)
printf("emitter current = %.3f mA \n",Ie)

clf()
x=linspace(0,18,100)
y=-(19.78/18)*x+19.78
xtitle("dc load line","Vce(volts)","Ic(mA)")
plot2d(x,y,style=3,rect=[0,0,19,20])
```

Scilab code Exa 16.17 voltage gain

```
//chapter16
//example16.17
//page357

Vcc=10 // V
R1= 10 // kilo ohm
R2=10 // kilo ohm
Vbe=0.7 // V
```

```
Re=5000 // ohm

V2=Vcc*R2/(R1+R2)
Ve=V2-Vbe
Ie=Ve/(Re/1000) // in mA
re_dash=25/Ie
Av=Re/(re_dash+Re)

printf("voltage gain = %.3f \n",Av)
```

Scilab code Exa 16.18 voltage gain

```
//chapter16
//example16.18
//page358

Re=5d3 // ohm
Rl=5d3 // ohm
re_dash=29.1 // in ohm, from example 16_17

Re_dash=Re*Rl/(Re+Rl)
Av=Re_dash/(re_dash+Re_dash)

printf("voltage gain = %.3f \n",Av)
```

Scilab code Exa 16.19 input impedence

```
//chapter16
//example16.19
//page359

Vcc=10 // V
R1= 10 // kilo ohm
```

```
R2=10 // kilo ohm
Vbe=0.7 // V
Re=4.3 // kilo ohm
gain_beta=200

V2=Vcc*R2/(R1+R2)
Ve=V2-Vbe
Ie=Ve/Re
re_dash=25/Ie
Re_dash=Re*R1/(Re+R1)
Zin_base=gain_beta*(re_dash+Re_dash)
Zin=Zin_base*(R1*R2/(R1+R2))/(Zin_base+R1*R2/(R1+R2)
    )

printf("input impedence = %.3f kilo ohm \n",Zin)

// the accurate answer is 4.996 kilo ohm but in book
    it is given as 4.96 kilo ohm
```

Scilab code Exa 16.20 output impedence

```
//chapter16
//example16.20
//page361

R1=3d3 // ohm
R2=4.7d3 // ohm
Rs=600 // ohm
re_dash=20// ohm
gain_beta=200

Rin_dash=R1*(R2*Rs/(R2+Rs))/(R1+(R2*Rs/(R2+Rs)))

Zout=re_dash+Rin_dash/gain_beta

```

```
printf("output impedence = %.1f ohm \n",Zout)
```

Chapter 17

Sinusoidal oscillations

Scilab code Exa 17.1 frequency of oscillations

```
//chapter17
//example17.1
//page375

L1=58.6d-6 // H
C1=300d-12 // F

f=1/(2*%pi*(L1*C1)^0.5)
printf("frequency of oscillations = %.3f Hz or %.3f
   kHz",f,f/1000)

// in book the answer is 1199 kHz but the accurate
   answer is 1200.358 kHz
```

Scilab code Exa 17.2 operating frequency

```
//chapter17
//example17.2
```

```
//page377

C1=0.001d-6 // F
C2=0.01d-6 // F
L=15d-6 // H

Ct=C1*C2/(C1+C2) // since both are in series

f=1/(2*%pi*(L*Ct)^0.5)
mv=C1/C2

printf("operating frequency = %.3f Hz or %.3f kHz \n
    ",f,f/1000)
printf("feedback function = %.3f",mv)

//in book the answer given is 1361 kHz but accurate
    answer is 1362.922 kHz
```

Scilab code Exa 17.3 feedback function

```
//chapter17
//example17.3
//page379

L1=1000d-6 // H
L2=100d-6 // H
M=20d-6 // H
C=20d-12 // F

Lt=L1+L2+2*M

f=1/(2*%pi*(Lt*C)^0.5)
mv=L2/L1

printf("operating frequency = %.3f Hz or %.3f kHz \n
```

```
",f,f/1000)
printf("feedback function = %.3f",mv)

//in book the answer is 1052 kHz but the accurate
   answer is 1054.029 kHz
```

Scilab code Exa 17.4 frequency of oscillations

```
//chapter17
//example17.4
//page381

R=1d6 // ohm
C=68d-12 // F

fo=1/(2*%pi*R*C*(6)^0.5)
printf("frequency of oscillations = %.3f Hz",fo)

// in book the answer given is 954 Hz but the
   accurate answer is 955.511 Hz
```

Scilab code Exa 17.5 frequency of oscillations

```
//chapter17
//example17.5
//page382

R=220d3 // ohm
C=250d-12 // F

f=1/(2*%pi*R*C)
printf("frequency of oscillations = %.3f Hz",f)

```

```
//in book the answer given is 2892 Hz but the
   accurate answer is 2893.726 Hz
```

Scilab code Exa 17.6 thickness of crystal

```
//chapter17
//example17.6
//page387

// frequency is inversely proportional to thickness
// so if thickness is reduced by 1%, frequency
   increases by 1%

printf("If thickness of crystal is reduced by 1
   percent, then \nfrequency is increased by 1
   percent \nbecause frequency is inversely
   proportional to thickness \n")
```

Scilab code Exa 17.7 resonant frequency

```
//chapter17
//example17.7
//page387

L=1 // H
C=0.01d-12 // F
Cm=20d-12 // F

fs=1/(2*%pi*(L*C)^0.5)
Ct=C*Cm/(C+Cm)
fp=1/(2*%pi*(L*Ct)^0.5)

```

```
printf("series resonant frequency = %.3f Hz  or %.3f kHz\n",fs,fs/1000)
printf("parallel resonant frequency = %.3f Hz or %.3f kHz\n",fp,fp/1000)

// in book the answer given is 1589 kHz for series resonant frequency but the accurate answer is 1591.549 kHz
// in book the answer given is 1590 kHz for parallel resonant frequency but the accurate answer is 1591.947 kHz
```

Chapter 18

Transistor tuned amplifiers

Scilab code Exa 18.1 impedence of circuit at resonance

```
//chapter18
//example18.1
//page396
L=1.25d-3 // H
C=250d-12 // F
R=10 // ohm

fr=(((1/(L*C))-(R^2/L^2))^0.5)/(2*%pi)
Zr=L/(C*R)
Q=2*%pi*fr*L/R

printf("resonant frequency of circuit = %.3f Hz or %
    .3f kHz \n",fr,fr/1000)
printf("impedence of circuit at resonance = %.3f ohm
     or %.3f kilo ohm \n",Zr,Zr/1000)
printf("quality factor of the circuit = %.3f",Q)
```

Scilab code Exa 18.2 circuit

```
//chapter18
//example18.2
//page396

L=100d-6 // H
C=100d-12 // F
R=10 // ohm
V=10 // V

fr=(((1/(L*C))-(R^2/L^2))^0.5)/(2*%pi)
Zr=L/(C*R)
I=V/Zr

printf("resonant frequency of circuit = %.3f Hz or %
    .3f kHz \n",fr,fr/1000)
printf("impedence of circuit at resonance = %.3f ohm
     or %.3f kilo ohm or %.3f mega ohm\n",Zr,Zr/1000,
    Zr/1d6)
printf("line current = %.4f ampere or %.3f micro
    ampere",I,I*1d6)

// the accurate answer for resonant frequency is
    1591.470 kHz
```

Scilab code Exa 18.3 frequency

```
//chapter18
//example17.3
//page398

fr=1200 // kHz
Q=60

BW=fr/Q
f1=fr-(BW/2)
```

```
f2=fr+(BW/2)

printf("bandwidth = %.3f kHz \n",BW)
printf("lower cut-off frequency = %.3f kHz \n",f1)
printf("upper cut-off frequency = %.3f kHz \n",f2)
```

Scilab code Exa 18.4 frequency

```
//chapter18
//example18.4
//page401

L=33d-3 // H
C=0.1d-6 // F
R=25 // ohm

fr=1/(2*%pi*(L*C)^0.5)
Xl=2*%pi*fr*L
Q=Xl/R
BW=fr/Q

printf("resonant frequency = %.3f Hz or %.3f kHz \n"
    ,fr,fr/1000)
printf("quality factor = %.3f \n",Q)
printf("bandwidth = %.3f Hz \n",BW)

// the accurate answer for bandwidth is 120.572 Hz
    but in book it is given as 120 Hz
// the accurate answer for quality factor is 22.978
    but in book it is given as 23
```

Scilab code Exa 18.5 co-efficient of coupling

```
//chapter18
//example18.5
//page402

BW=200 // kHz
fr=10d3 // kHz

k=BW/fr

printf("co-efficient of coupling = %.3f \n",k)
```

Scilab code Exa 18.6 ac load

```
//chapter18
//example18.6
//page405

L=50.7d-6 // H
C=500d-12 // F
R=10 // ohm
Rl=1d6 // ohm

fr=1/(2*%pi*(L*C)^0.5)
R_dc=R
Xl=2*%pi*fr*L
Q=Xl/R
Rp=Q*Xl
R_ac=Rp*Rl/(Rp+Rl)

printf("resonant frequency = %.3f Hz or %.3f kHz \n"
    ,fr,fr/1000) // amswer in book is incorrect
printf("dc load = %.3f ohm \n",R_dc)
printf("ac load = %.3f ohm or %.3f kilo ohm \n",R_ac
    ,R_ac/1000)

```

```
// in book the aswer for resonant frequency is 106
   Hz which is incorrect
// the correct answer is 999.611 kHz

// the accurate answer for ac load is 10.038 kilo
   ohm
```

Scilab code Exa 18.7 load

```
//chapter18
//example18.7
//page406

Vcc=50 // V
Np=5
Ns=1
R=50 // ohm
R_ac=(Np/Ns)^2*R
Po=Vcc^2/R_ac

printf("ac load = %.3f ohm \n",R_ac)
printf("maximum load power = %.3f W \n",Po)
```

Chapter 19

Modulation and demodulation

Scilab code Exa 19.1 Vmax&Vmin

```
//chapter19
//example19.1
//page416

// the figure in book is for reference only as
   equations for Ec and Es are already explained in
   the theory in the book.

printf("Ec=(Vmax+Vmin)/2 \n")
printf("Es=(Vmax-Vmin)/2 \n")
printf("But, Es=m*Ec \n")
printf("So (Vmax-Vmin)/2 = m*(Vmax+Vmin)/2 \n")
printf("thus, m = (Vmax-Vmin)/(Vmax+Vmin) \n")
```

Scilab code Exa 19.2 modulation factor

```
//chapter19
//example19.2
```

```
//page416

// figure is given in book for understanding purpose
    only.It is not required for solving the example
   as maximum and minimum peak voltages are given in
    the problem statement itself.

Vmax_pp=16 // mV
Vmin_pp=4 // mV

Vmax=Vmax_pp/2
Vmin=Vmin_pp/2

m=(Vmax-Vmin)/(Vmax+Vmin)

printf("modulation factor = %.3f \n",m)
```

Scilab code Exa 19.3 modulation factor

```
//chapter19
//example19.3
//page417

Es=50 // V
Ec=100 // V

m=Es/Ec

printf("modulation factor = %.3f \n",m)
```

Scilab code Exa 19.4 Bandwidth for RF amplifier

```
//chapter19
```

```
//example19.4
//page419

fc=2500 // kHz
fs_min=0.05 // kHz
fs_max=15 // kHz

upper_sideband_min=fc+fs_min
upper_sideband_max=fc+fs_max

lower_sideband_min=fc-fs_min
lower_sideband_max=fc-fs_max

BW=upper_sideband_max-lower_sideband_max

printf("lower sideband is from %.3f to %.3f kHz \n",lower_sideband_min,lower_sideband_max)
printf("upper sideband is from %.3f to %.3f kHz \n",upper_sideband_min,upper_sideband_max)
printf("Bandwidth for RF amplifier = %.3f kHz \n",BW)
```

Scilab code Exa 19.5 amplitudes of components

```
//chapter19
//example19.5
//page420

// v=5*(1+0.6*cos(6280*t))*sin(211d4*t) V
// compare with v=Ec*(1+m*cos(ws*t))*sin(wc*t) we get
Ec=5 // V
m=0.6
fs=6280/(2*%pi) // Hz
fc=211d4/(2*%pi) // Hz
```

```

Vmin=Ec-m*Ec
Vmax=Ec+m*Ec

f1=(fc-fs)/1000 // in kHz
f2=fc/1000 // in kHz
f3=(fc+fs)/1000 // in kHz

V1=m*Ec/2
V2=Ec
V3=m*Ec/2

printf("minimum amplitude = %.3f V and maximum amplitude = %.3f V \n",Vmin,Vmax)
printf("frequency components = %.1f kHz, %.1f Hz, %.1fkHz \n",f1,f2,f3)
printf("amplitudes of components = %.3f V, %.3f V, %.3f V \n",V1,V2,V3)

// in book there is error of 0.2 kHz in every frequency component. The accurate answers are 334.8,335.8,336.8 kHz
```

Scilab code Exa 19.6 amplitude of each sideband term

```
//chapter19
//example19.6
//page420

fc=1000 // kHz
fs=5 // kHz
m=0.5
Ec=100 // V

lower_sideband=fc-fs
```

```
upper_sideband=fc+fs
amplitude=m*Ec/2

printf("lower and upper sideband frequencies = %.3f
   kHz and %.3f kHz \n",lower_sideband,
   upper_sideband)
printf("amplitude of each sideband term = %.3f V \n"
   ,amplitude)
```

Scilab code Exa 19.7 power of modulated wave

```
//chapter19
//example19.7
//page422

Pc=500 // W
m=1

Ps=0.5*m^2*Pc
Pt=Pc+Ps

printf("sideband power = %.3f W \n",Ps)
printf("power of modulated wave = %.3f W \n",Pt)
```

Scilab code Exa 19.8 sideband power

```
//chapter19
//example19.8
//page422

m1=0.8
m2=0.1
Pc=50 // kW
```

```
Ps1=0.5*m1^2*Pc
Ps2=0.5*m2^2*Pc

printf("for m=0.8, sideband power = %.3f kW \n",Ps1)
printf("for m=0.1, sideband power = %.3f kW \n",Ps2)
```

Scilab code Exa 19.9 modulating signal

```
//chapter19
//example19.9
//page422

// block diagram is for understanding purpose inly.
   It is not required to solve the example
m=1
eta=0.72
// carrier is not affected by modulating signal so
   its power level remains unchanged before and
   after modulation
Pc=40 // kW
Ps=0.5*m^2*Pc
P_audio=Ps/eta

printf("carrier power after modulation = %.3f kW \n"
   ,Pc)
printf("required audio power = %.3f kW \n",P_audio)
```

Scilab code Exa 19.10 frequencie & bandwith

```
//chapter19
//example19.10
//page423
```

```
fc=500 // kHz
fs=1 // kHz

lower_sideband=fc-fs
upper_sideband=fc+fs
BW=upper_sideband-lower_sideband

printf("sideband frequencies = %.3f kHz and %.3f kHz
    \n",lower_sideband,upper_sideband)
printf("bandwidth required = %.3f kHz \n",BW)
```

Scilab code Exa 19.11 antenna current

```
//chapter19
//example19.11
//page423

m=0.4
Ic=8 // A
// Pt=Pc+Ps and Ps=0.5*m^2*Pc so Pt=Pc*(1+m^2/2)
// so Pt/Pc=1+m^2/2 but P is proportional to I^2 so
// (It/Ic)^2=1+m^2/2 and thus we get

It=Ic*(1+m^2/2)^0.5

printf("antenna current for m=0.4 is = %.3f A \n",It
   )
```

Scilab code Exa 19.12 modulation factor

```
//chapter19
//example19.12
```

```
//page424

It=8.93 // A
Ic=8 // A

// we know that (It/Ic)^2=1+m^2/2 so making m as
   subject we get
m=(2*((It/Ic)^2-1))^0.5

printf("modulation factor = %.3f or %.3f percent \n"
   ,m,m*100)
```

Scilab code Exa 19.13 modulation factor

```
//chapter19
//example19.13
//page424

Vt=110 // V
Vc=100 // V

// since Pt/Pc=1+m^2/2 and P is proportional to V^2
   we get (Vt/Vc)^2=1+m^2/2
// making m as subject we get

m=(2*((Vt/Vc)^2-1))^0.5

printf("modulation factor = %.3f or %.3f percent \n"
   ,m,m*100)
```

Scilab code Exa 19.14 AM wave

```
//chapter19
```

```
//example19.14
//page424

Vc=5 // V
V_lower=2.5 // V
V_upper=2.5 // V
R=2 // kilo ohm

// figure given in book is just for understanding
    purpose.It is not a part of solution.
// however, the figure has been made in xcos and
    screenshot has been attached for reference

// since power=(rms voltage)^2/R we get

Pc=(0.707*Vc)^2/R
P_lower=(0.707*V_lower)^2/R
P_upper=(0.707*V_upper)^2/R
Pt=Pc+P_lower+P_upper

printf("power delivered by carrier = %.3f mW \n",Pc)
printf("power delivered by lower sideband = %.3f mW
    \n",P_lower)
printf("power delivered by upper sideband = %.3f mW
    \n",P_upper)
printf("total power delivered by AM wave = %.3f mW \
    n",Pt)
```

Chapter 20

Regulated dc power supply

Scilab code Exa 20.1 regulation

```
//chapter20
//example20.1
//page437

V_NL=400 // V
V_FL=300 // V

regulation=((V_NL-V_FL)/V_FL)*100

printf("percent voltage regulation = %.3f percent \n
    ",regulation)
```

Scilab code Exa 20.2 full load voltage

```
//chapter20
//example20.2
//page437

```

```
V_NL=30 // V
regulation=1

// since regulation=((V_NL-V_FL)/V_FL)*100, we get V_FL as

V_FL=100*V_NL/(100+regulation)
printf("full load voltage = %.3f V \n",V_FL)
```

Scilab code Exa 20.3 regulation for power

```
//chapter20
//example20.3
//page437

// for power supply A
V_NL1=30 // V
V_FL1=25 // V

regulation1=((V_NL1-V_FL1)/V_FL1)*100

//for power supply B
V_NL2=30 // V
V_FL2=29 // V

regulation2=((V_NL2-V_FL2)/V_FL2)*100

printf("regulation for power supply A =%.3f percent \n",regulation1)
printf("regulation for power supply B =%.3f percent \n",regulation2)

if regulation1>regulation2 then
        printf("thus, power supply B is better \n")
    elseif regulation2>regulation1 then
```

```
        printf("thus, power supply A is better \n")
    else    printf("both are equally good \n")
end
```

Scilab code Exa 20.4 minimum load resistance

```
//chapter20
//example20.4
//page438

V_NL=500 // V
V_FL=300 // V
I_FL=120 // mA

regulation=((V_NL-V_FL)/V_FL)*100

Rl_min=V_FL/I_FL

printf("voltage regulation = %.3f percent \n",
   regulation)
printf("minimum load resistance = %.3f kilo ohm \n",
   Rl_min)
```

Scilab code Exa 20.5 maximum zener current

```
//chapter20
//example20.5
//page441

Vin=24 // V
Vout=12 // V
Rs=160 // ohm
Rl_min=200 // ohm
```

```

Is=(Vin-Vout)/Rs  // in ampere

// minimum load occurs when Rl tends to infinity so
Il_min=0

// maximum load occurs when Rl=200 ohm
Il_max=Vout/Rl_min  // in ampere

Iz_min=Is-Il_max  // in ampere
Iz_max=Is-Il_min  // in ampere

printf("current through series reistance = %.3f mA \n \n",Is*1000)
printf("minimum load current = %.3f mA \n",Il_min*1000)
printf("maximum load current = %.3f mA \n",Il_max*1000)
printf("minimum zener current = %.3f mA \n",Iz_min*1000)
printf("maximum zener current = %.3f mA \n \n",Iz_max*1000)

printf("comment : current Is through Rs is constant.\nAs load current increases from 0 to 60 mA, zener current decreases from 75 to 15 mA, \nmaintaining Is constant.\nThis is the normal operation of zener regulator \ni.e.Is and Vout remain constant inspite of changes in load or source voltage.")
```

Scilab code Exa 20.6 required series resistance

```
//chapter20
//example20.6
```

```
//page441

Vin_min=22 // V
Vout=15 // V
Il_max=0.1 // A

// for maximum series resistance, we consider the
   case when input voltage is minimum and load
   current is maximum because then zener current
   drops to minimum.Thus,
Rs_max=(Vin_min-Vout)/Il_max

printf("required series resistance = %.3f ohm \n",
   Rs_max)
```

Scilab code Exa 20.7 load voltage & current

```
//chapter20
//example20.7
//page442

Vz=10 // V
Vbe=0.5 // V
Rl=1000 // ohm

Vout=Vz-Vbe
Il=Vout/Rl

printf("load voltage = %.3f V \n",Vout)
printf("load current = %.3f mA \n",Il*1000)
```

Scilab code Exa 20.8 Zener diode

```
//chapter20
//example20.8
//page441

Ic=1 // A
gain=50
Vout=6 // V
Vbe=0.5 // V
Vin=10 // V
Iz=10d-3 // A

Ib=Ic/gain
Vz=Vbe+Vout // Vout=Vz-Vbe

V_Rs=Vin-Vz
Rs=V_Rs/(Ib+Iz)

printf("required breakdown voltage for zener diode =
    %.3f V \n",Vz)
printf("required value of Rs = %.3f ohm \n",Rs)

// in book Rs=117 ohm but accurate answer is 116.667
    ohm

// note : in xcos, there is no Zener diode so in the
    result (circuit) file a simple diode is used to
   represent a zener diode
```

Scilab code Exa 20.9 Zener diode

```
//chapter20
//example20.9
//page443

Vz=12 // V
```

```
Vbe=0.7 // V
Vin=20 // V
Rs=220 // ohm
Rl=1d3 // ohm
gain=50

Vout=Vz-Vbe
V_Rs=Vin-Vz
I_Rs=V_Rs/Rs
Il=Vout/Rl
Ic=Il
Ib=Ic/gain
Iz=I_Rs-Ib

printf("output voltage = %.3f V \n",Vout)
printf("zener current = %.3f mA \n",Iz*1000)
```

Scilab code Exa 20.10 regulated output voltage

```
//chapter20
//example20.10
//page445

R2=1 // kilo ohm
R1=2 // kilo ohm
Vz=6 // V
Vbe=0.7 // V

m=R2/(R1+R2)
A_CL=1/m
Vout=A_CL*(Vz+Vbe)

printf("regulated output voltage = %.3f V \n",Vout)
```

Scilab code Exa 20.11 closed loop voltage gain

```
//chapter20
//example20.11
//page445

R2=10 // kilo ohm
R1=30 // kilo ohm

m=R2/(R1+R2)
A_CL=1/m

printf("closed loop voltage gain = %.3f \n",A_CL)
```

Scilab code Exa 20.12 collector current

```
//chapter20
//example20.12
//page446

Vz=8.3 // V
Vbe=0.7 // V
Rl=100 // ohm
Rs=130 // ohm
Vin=22 // V

Vout=Vz+Vbe
Il=Vout/Rl
Is=(Vin-Vout)/Rs
Ic=Is-Il

printf("regulated output voltage = %.3f V \n",Vout)
```

```
printf("load current = %.3f mA \n",Il*1000)
printf("current through Rs = %.3f mA \n",Is*1000)
printf("collector current = %.3f mA \n",Ic*1000)
```

Chapter 21

Solid state switching circuits

Scilab code Exa 21.1 voltage required to saturate transistor

```
//chapter21
//example21.1
//page456

Vcc=10 // V
Vbe=0.7 // V
Rb=47d3 // ohm
Rc=1d3 // ohm
gain=100

Ic_sat=Vcc/Rc
Ib=Ic_sat/gain
V_plus=Ib*Rb+Vbe

printf("voltage required to saturate transistor = +%
   .3f V \n",V_plus)
```

Scilab code Exa 21.2 square wave

```
//chapter21
//example21.2
//page463

R=10d3 // ohm
C=0.01d-6 // F

T=1.4*R*C
f=1/T

printf("time period of square wave = %.3f ms \n",T
    *1000)
printf("frequency of square wave = %.3f kHz \n",f
    /1000)

// the accurate answer for frequency is 7.143 kHz
    but in book it is given 7 kHz
```

Scilab code Exa 21.3 differentiating circuit

```
// chapter 21
// example 21.3
// page 468

printf("In RC differentiating circuit, the output
    votage is taken across \nR and waveform of output
     depends on time constant of \ncircuit. For
    proper functioning, product RC should be many \
    ntimes smaller than time period of input wave. \n
    ")
```

Scilab code Exa 21.4 output voltage

```
//chapter21
//example21.4
//page468

R=10d3 // ohm
C=2.2d-6 // F
V1=0 // V
V2=10 // V
t1=0 // sec
t2=0.4 // sec

Eo=R*C*(V2-V1)/(t2-t1)

printf("output voltage = %.3f V \n",Eo)
```

Scilab code Exa 21.5 plotting input and output waveforms

```
//chapter21
//example21.5
//page472

Vin_peak=12 // V

// for positive half cycle diode conducts so
Vout_peak=Vin_peak-0.7 // V

// for negative half cycle diode does not conduct so
Vout_min=0 // V

printf("peak output voltage = %.3f V in positive
    half cycle and \n                    %.3f V in
     negative half cycle",Vout_peak,Vout_min)

// plotting input and output waveforms in same graph
     using following code instead of using xcos
```

```
clf()
t=linspace(0,2*%pi,100)
Vin=12*sin(t)
Vout=Vout_peak*sin(t)+Vout_min
plot2d(t,Vin,style=2,rect=[0,0,10,20])
xtitle("input - blue     output - green","t","volts"
  )
plot2d(t,Vout,style=3,rect=[0,0,10,20])
```

Scilab code Exa 21.6 plotting input and output waveforms

```
//chapter21
//example21.6
//page472

Rl=4 // kilo ohm
R=1 // kilo ohm
Vin_peak=10 // V

Vout_peak=Vin_peak*Rl/(Rl+R)
Vout_min=0 // because of diode
printf("peak output voltage = %.3f V \n",Vout_peak)

// plotting input and output waveforms in same graph
   using following code instead of using xcos
clf()
t=linspace(0,2*%pi,100)
Vin=Vin_peak*sin(t)
Vout=Vout_peak*sin(t)+Vout_min
plot2d(t,Vin,style=2,rect=[0,0,10,20])
xtitle("input - blue     output - green","t","volts"
  )
plot2d(t,Vout,style=3,rect=[0,0,10,20])
```

Scilab code Exa 21.7 plotting input and output waveforms

```
//chapter21
//example21.7
//page473

V=-10 // V
Vout=-0.7 // V

Vr=V-Vout

printf("output voltage = %.3f V \n",Vout)
printf("voltage across R = %.3f V \n",Vr)

// plotting input and output waveforms in same graph
    using following code instead of using xcos
clf()
t=linspace(0,%pi,100)
Vin=V*sin(t)
Vout=Vr*sin(t)
subplot(1,2,2)
plot2d(t,Vout,style=3,rect=[0,-0.7,10,11])
xtitle("Vout","t","volts")

subplot(1,2,1)
plot2d(t,Vin,style=2,rect=[0,-11,10,1])
xtitle("Vin","t","volts")
```

Scilab code Exa 21.8 plotting input and output waveforms

```
//chapter21
//example21.8
```

```
//page473

Rl=1d3 // ohm
R=200 // ohm

// for positive half cycle, diode is forward biased
    and since load is in parallel with diode we get
V_out_p=0.7 // V

// for negative half cycle, diode is reverse biased
    so it is open. Hence
V_in=-10 // V
V_out_n=V_in*Rl/(Rl+R)

printf("output voltage for positive cycle = %.3f V \
    nand for negative cycle = %.3f V",V_out_p,V_out_n
    )

// plotting input and output waveforms in same graph
     using following code instead of using xcos
clf()
t=linspace(0,%pi,100)
Vin=V_in*sin(t)
Vout=-V_out_n*sin(t)
subplot(2,2,1)
plot2d(t,-Vin,style=3,rect=[0,0,10,11])
xtitle("Vin +ve","t","volts")
subplot(2,2,2)
plot2d(t,Vout,style=2,rect=[0,-5,10,0.7])
xtitle("Vout","t","volts")
t=linspace(%pi,2*%pi,100)
Vin=V_in*sin(t)
subplot(2,2,3)
plot2d(t,-Vin,style=3,rect=[0,-11,10,0])
xtitle("Vin -ve","t","volts")
subplot(2,2,4)
plot2d(t,-Vout,style=2,rect=[0,-11,10,0])
xtitle("Vout","t","volts")
```

Scilab code Exa 21.9 empty

```
//chapter21
//example21.9
//page474

printf("The purpose of using series resistance R is
   : \n 1) if R is not present, diode will short
   voltage source in positive half cycle \n 2) so
   large current will flow which may damage voltage
   source or diode. \n To prevent this i.e. to
   protect diode and voltage source, R is used.")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 21.10 plot input and output waveforms

```
// chapter 21
// example 21.10
// page 478

V=2 // V
Vin=5 // V

// during positive half cycle
Vc_p=Vin-V // since Vin-Vc-V=0
// thus capacitor charges to Vc_p

// during negative half cycle
Vout=-Vin-Vc_p // since Vin-Vc_p-Vout=0
```

```

// we plot input and output waveforms using the
   following code instead of using xcos

clf()
t=0:0.1:5*%pi
plot(t,5*squarewave(t,50))
plot2d(t,-Vc_p+(-Vout+V)*squarewave(t,50)/2,style=3)
xtitle("input - blue                    output -
   green","t","volts")
```

Scilab code Exa 21.11 plot input and output waveforms

```
// chapter 21
// example 21.11
// page 479

V=-2 // V
Vin=5 // V

// during positive half cycle
Vc_p=Vin-V // since Vin-Vc-V=0
// thus capacitor charges to Vc_p

// during negative half cycle
Vout=-Vin-Vc_p // since Vin-Vc_p-Vout=0

// we plot input and output waveforms using the
   following code instead of using xcos

clf()
t=0:0.1:5*%pi
plot(t,5*squarewave(t,50))
plot2d(t,-Vc_p+(-Vout+V)*squarewave(t,50)/2,style=3)
xtitle("input - blue                    output -
```

```
green","t","volts")
```

Chapter 22

Field Effect Transistors

Scilab code Exa 22.1 Id

```
//chpater22
//example22.1
//page491

I_DSS=12 // mA
V_GS_off=-5 // V

printf("I_D=%d*(1+V_GS/%d)^2 mA \n",I_DSS,-V_GS_off)
```

Scilab code Exa 22.2 drain current

```
//chapter22
//example22.2
//page491

I_DSS=32 // mA
V_GS=-4.5 // V
V_GS_off=-8 // V
```

```
I_D=I_DSS*(1-V_GS/V_GS_off)^2

printf("drain current = %.3f mA \n",I_D)
```

Scilab code Exa 22.3 pinch off voltage

```
//chapter22
//example22.3
//page491

I_D=5 // mA
I_DSS=10 // mA
V_GS_off=-6 // V

// we know that I_D=I_DSS*(1-V_GS/V_GS_off)^2 so
   making V_GS as subject we get

V_GS=V_GS_off*(1-(I_D/I_DSS)^0.5)
V_P=-V_GS_off

printf("gate source voltage = %.3f V \n",V_GS)
printf("pinch off voltage = %.3f V \n",V_P)
```

Scilab code Exa 22.4 gate source resistance

```
//chapter22
//example22.4
//page493

V_GS=15 // V
I_G=1d-9 // A

```

```
R_GS=V_GS/I_G

printf("gate source resistance = %.3f ohm or %.3f
   mega ohm \n",R_GS,R_GS/1d6)
```

Scilab code Exa 22.5 transconductance

```
//chapter22
//example22.5
//page493

Vgs1=-3.1 // V
Vgs2=-3 // V
Id1=1d-3 // A
Id2=1.3d-3 // A

g_fs=(Id2-Id1)/(Vgs2-Vgs1)

printf("transconductance = %.3f mho or %.3f micro
   mho \n",g_fs,g_fs*1d6)
```

Scilab code Exa 22.6 amplification factor

```
//chapter22
//example22.6
//page493

// for V_GS = 0V constant
V_DS1=7 // V
V_DS2=15 // V
I_D1=10 // mA
I_D2=10.25 // mA

```

```
rd=(V_DS2-V_DS1)/(I_D2-I_D1)

// for V_DS = 15V constant
V_GS1=0
V_GS2=0.2
I_D1=9.65
I_D2=10.25

g_fs=(I_D2-I_D1)/(V_GS2-V_GS1)

mu=rd*g_fs

printf("ac drain resistance = %.3f ohm or %.3f kilo
   ohm \n",rd/1000,rd)
printf("transconductance = %.3f mho or %.3f micro
   mho \n",g_fs,g_fs*1000)
printf("amplification factor = %.3f \n",mu)
```

Scilab code Exa 22.7 Kirchoff's law

```
//chapter22
//example22.7
//page496

I_DSS=5d-3 // A
V_DD=20 // V
V_DS=10 // V
V_P=-2 // V
V_G=0 // V
I_D=1.5d-3 // A

V_GS=V_P*(1-((I_D/I_DSS)^0.5)) // I_D=I_DSS*(1-V_GS/
   V_P)^2
V_S=V_G-V_GS
R_S=V_S/I_D
```

```
// by Kirchoff's law we get V_DD=I_D*R_D+V_DS+I_D*
   R_S so making R_D as subject we get
R_D=(V_DD-V_DS-I_D*R_S)/I_D

printf("Rs = %.3f kilo ohm and Rd = %.3f kilo ohm \n
   ",R_S/1000,R_D/1000)
```

Scilab code Exa 22.8 required value of Rs

```
//chapter22
//example22.8
//page496

V_P=-5 // V
V_DD=30 // V
I_DSS=10 // mA
I_D=2.5 // mA
R1=1000 // kilo ohm
R2=500 // kilo ohm

// since I_D=I_DSS*(1-(V_GS/V_P))^2, making V_GS as
   subject we get

V_GS=V_P*(1-(I_D/I_DSS)^0.5)

V2=V_DD*R2/(R1+R2)

// since V2 = V_GS + I_D*Rs, making Rs as subject we
   get

Rs=(V2-V_GS)/I_D

printf("required value of Rs = %.3f kilo ohm \n",Rs)
```

Scilab code Exa 22.9 voltage amplification of the circuit

```
//chapter22
//example22.9
//page497

R_L=10d3 // ohm
g_fs=3000d-6 // mho

// since rd >> R_L, we can write

Av=g_fs*R_L

printf("voltage amplification of the circuit = %.3f
   \n",Av)
```

Scilab code Exa 22.10 Vds

```
//chapter22
//example22.10
//page498

V_DD=30 // V
I_D=2.5d-3 // A
R_D=5d3 // ohm
R_S=200 // ohm

V_DS=V_DD-I_D*(R_D+R_S)
V_GS=-I_D*R_S

printf("V_DS = %.3f V \n",V_DS)
printf("V_GS = %.3f V \n",V_GS)
```

Scilab code Exa 22.11 dc voltage of drain

```
//chapter22
//example22.11
//page498

V_DD=30 // V
I_D1=2.15d-3 // A
I_D2=9.15d-3 // A
R_D1=8.2d3 // ohm
R_D2=2d3 // ohm
R_S1=680 // ohm
R_S2=220 // ohm

V_RD1=I_D1*R_D1
V_D1=V_DD-V_RD1
V_S1=I_D1*R_S1

V_RD2=I_D2*R_D2
V_D2=V_DD-V_RD2
V_S2=I_D2*R_S2

printf("For stage 1 : dc voltage of drain = %.3f V
    and source = %.3f V \n",V_D1,V_S1)
printf("For stage 2 : dc voltage of drain = %.3f V
    and source = %.3f V \n",V_D2,V_S2)
```

Chapter 23

Silicon controlled rectifiers

Scilab code Exa 23.1 Breakover voltage

```
//chapter23
//example23.1
//page509

printf("1) Breakover voltage of 400V : It means that
    if gate is open and the \n")
printf("   supply voltage is 400V, then SCR will
   start conducting heavily. \n")
printf("   However,as long as the supply voltage <
   400V, SCR stays open. \n \n")

printf("2) Trigger current of 10mA : It means that
   if the supply voltage is \n")
printf("   less than breakover voltage and a minimum
    gate current of 10 mA \n")
printf("   is passed, SCR conducts. It wont conduct
   if gate current is less \n")
printf("   than 10mA. \n \n")

printf("3) Holding current of 10mA : When SCR is
   conducting, it will not open \n")
```

```
printf("    even if triggering current is removed. However, if supply voltage \n")
printf("    is reduced, anode current also decreases. When anode current drops \n")
printf("    to 10mA, the holding current, SCR turns off. \n \n")

printf("4) If gate current is increased to 15mA, the SCR will be turned on \n")
printf("    lower supply voltage. \n")
```

Scilab code Exa 23.2 rating

```
//chapter23
//example23.2
//page510

t=12d-3 // sec
I=50 // A
fuse_rating=I^2*t

if fuse_rating < 90
     printf("rating = %.3f ampere square second which is less than maximum \nrating so device will not be destroyed \n",fuse_rating)
else printf("rating = %.3f ampere square second which is more than maximum \nrating so device may get damaged \n",fuse_rating)

end
```

Scilab code Exa 23.3 maximum allowable duration of surge

```
//chapter23
//example23.3
//page510

rating=50 // ampere square second
Is=100 // A

t_max=rating/Is^2

printf("maximum allowable duration of surge = %.3f s
    or %.3f ms\n",t_max,t_max*1000)
```

Scilab code Exa 23.4 average current

```
//chapter23
//example23.4
//page514

v=100 // V
Vm=200 // V
R_L=100 // ohm

// since v=Vm*sin(theta), we get

theta=asin(v/Vm)*180/%pi // in terms of degrees

phi=180-theta

V_avg=Vm*(1+cos(theta*%pi/180))/(2*%pi)

I_avg=V_avg/R_L

printf("firing angle = %.2f degrees \n",theta)
printf("conduction angle = %.2f degrees \n",phi)
printf("average current = %.4f A \n",I_avg)
```

```

// the accurate answer for average current is 0.594
    A but in book it is given as 0.5925 A
```

Scilab code Exa 23.5 power output

```
//chapter23
//example23.5
//page515

Vm=400 // V
v=150 // V
R_L=200 // ohm

// since v=Vm*sin(theta), we get

theta=asin(v/Vm)*180/%pi // in terms of degrees

V_av=Vm*(1+cos(theta*%pi/180))/(2*%pi)
I_av=V_av/R_L
P=V_av*I_av

printf("firing angle = %.2f degrees \n",theta)
printf("average output voltage = %.3f V \n",V_av)
printf("average current for load of 200 ohm = %.3f A
     \n",I_av)
printf("power output = %.3f W \n",P)

// the accurate answer for power output is 75.250 W
    but in book it is given as 75.15 W
```

Scilab code Exa 23.6 forward breakdown voltage

```
//chapter23
//example23.6
//page515

Vm=240 // V
v=180 // V

// figure given is for understanding purpose only.
   It is not required to solve the example

// SCR remains off till it reaches 180 V i.e.
   forward breakdown voltage

// since v=Vm*sin(theta), we get

theta=asin(v/Vm) // firing angle in terms of degrees

// since theta=314*t, we get

t=theta/314 // seconds

printf("off duration of SCR = %.3f ms \n",t*1000) //
   multiply t by 1000 to display time in
   milliseconds
```

Scilab code Exa 23.7 voltage

```
//chapter23
//example23.7
//page517

alpha=60 // degrees
Vm=200 // V
R_L=100 // ohm

```

```
V_av=Vm*(1+cos(alpha*%pi/180))/%pi

I_av=V_av/R_L

printf("dc output voltage = %.3f V \n",V_av)
printf("load current for firing angle of 60 degrees
   = %.3f A \n",I_av)
```

Chapter 24

Power electronics

Scilab code Exa 24.1 (RB1+RB2) = RB1/Rbb

```
//chapter24
//example24.1
//page533

RBB=10 // kilo ohm
eta=0.6

//eta=RB1/(RB1+RB2) = RB1/Rbb so
RB1=eta*RBB
RB2=RBB-RB1
printf("RB1 = %.3f kilo ohm \n",RB1)
printf("RB2 = %.3f kilo ohm",RB2)
```

Scilab code Exa 24.2 voltage

```
//chapter24
//example24.2
//page533
```

```

VBB=10 // V
eta=0.65
VD=0.7 // V

stand_off_voltage=eta*VBB
peak_point_voltage=VD+eta*VBB

printf("stand off voltage = %.3f V \n",
   stand_off_voltage)
printf("peak point voltage = %.3f V \n",
   peak_point_voltage)
```

Chapter 25

Electronic instruments

Scilab code Exa 25.1 sensitivity

```
//chapter25
//example25.1
//page543

Ig=1d-3 // A

S=1/Ig

printf("sensitivity = %.3f ohm per volt \n",S)
```

Scilab code Exa 25.2 equivalent resistance of meter

```
//chapter25
//example25.2
//page543

S=1000 // ohm per volt
V=50 // V
```

```
R=50d3 // ohm

R_meter=S*V

R_equi=R*R_meter/(R+R_meter) //equivalent resistance of meter and given resistance across which meter is connected

printf("ratio of circuit resistance before and after connecting multimeter = %.3f \n",R/R_equi)
printf("Thus equivalent resistance is reduced to half. So current drawn is double \n")
printf("Thus multimeter will give highly incorrect reading \n \n")
printf("As a rule, multimeter resistance should be 100 times the resistance across \nwhich voltage is to be measured \n")
```

Scilab code Exa 25.3 voltage read by multimeter

```
//chapter25
//example25.3
//page544

S=4 // kilo ohm per volt
V_range=10 // V
V=20 // V
R=10 // kilo ohm

R_meter=S*V_range
R_equi=R+R*R_meter/(R+R_meter)
I=V/R_equi
V_reading=I*R*R_meter/(R+R_meter)

printf("voltage read by multimeter = %.3f V \n",
```

```
V_reading)
```

Scilab code Exa 25.4 voltage read by multimete

```
//chapter25
//example25.4
//page544

S=20 // kilo ohm per volt
V_range=10 // V
V=20 // V
R=10 // kilo ohm

R_meter=S*V_range
R_equi=R+R*R_meter/(R+R_meter)
I=V/R_equi
V_reading=I*R*R_meter/(R+R_meter)

printf("voltage read by multimeter = %.3f V \n",
    V_reading)

// answer in book is 9.88V but accurate answer is
    9.756V
```

Scilab code Exa 25.5 resistance of voltmeter

```
//chapter25
//example25.5
//page545

R1=20d3 // ohm
R2=20d3 // ohm
R3=30d3 // ohm
```

```
R4=30d3 // ohm
V=100 // V
Rm=60d3 // ohm

// case 1 : meter is not connected
R=R1+R2+R3+R4
I=V/R
V_A=V
V_B=V-I*R2
V_C=V-I*(R1+R2)
V_D=V-I*(R1+R2+R3)

// case2 : meter is connected
    // At A
    V_A1=V

    // At B
    R_total_B=R1+(Rm*(R2+R3+R4)/(Rm+R2+R3+R4))
    I1=V/R_total_B
    V_B1=I1*(Rm*(R2+R3+R4)/(Rm+R2+R3+R4))

    // At C
    R_total_C=R1+R2+(Rm*(R3+R4)/(Rm+R3+R4))
    I2=V/R_total_C
    V_C1=V*(Rm*(R3+R4)/(Rm+R3+R4))/R_total_C

    // At D
    R_total_D=R1+R2+R3+(Rm*R4/(Rm+R4))
    I2=V/R_total_D
    V_D1=V*(Rm*R4/(Rm+R4))/R_total_D

printf("CASE 1 : meter is not connected \n      Voltage at A = %.3f V \n      Volatge at B = %.3f V \n      Volatge at C = %.3f V \n      Volatge at D = %.3f V \n",V_A,V_B,V_C,V_D)
printf("CASE 2 : meter is connected \n      At A then voltage at A = %.3f V",V_A1)
printf("\n      At B then voltage at B = %.3f V",
```

```
V_B1)
printf("\n        At C then voltage at C = %.3f V",
   V_C1)
printf("\n        At D then voltage at D = %.3f V \n \
   n",V_D1)

printf("resistance of voltmeter should be 100 times
   the resistance across \nwhich voltage is to be
   measured.Since this condition is not \nsatisfied
   here, readings are wrong. \n")
```

Scilab code Exa 25.6 spot shift

```
//chapter25
//example25.6
//page552

S=0.01 //mm per volt
V=400 // V

spot_shift=S*V

printf("spot shift = %.3f mm \n",spot_shift)
```

Scilab code Exa 25.7 applied voltage

```
//chapter25
//example25.7
//page552

S=0.03 // mm per volt
spot_shift=3 // mm

```

```
V=spot_shift/S // since spot shift = deflection
    sensitivity * applied voltage

printf("applied voltage = %.3f V \n",V)
```

Scilab code Exa 25.8 unknown voltage

```
//chapter25
//example25.8
//page555

V1=200 // V
d1=2 // cm
d2=3 // cm

// since sensitivity = voltage / deflcetion we get
S=V1/d1
V2=S*d2

printf("unknown voltage = %.3f V",V2)
```

Scilab code Exa 25.9 ratio of fv to fh

```
//chapter25
//example25.9
//page556

fh=1000 // Hz

// case (i) :- ratio of fv to fh = 1:1
fv1=1*fh

// case (ii) :- ratio = 2:1
```

```
fv2=2*fh

// case (iii) :- ratio = 6:1
fv3=6*fh

printf("for case1 i.e. fv/fh = 1/1, fv = %.3f Hz \n",fv1)
printf("for case2 i.e. fv/fh = 2/1, fv = %.3f Hz \n",fv2)
printf("for case3 i.e. fv/fh = 6/1, fv = %.3f Hz \n",fv3)
```

Chapter 26

Integrated circuits

Scilab code Exa 26.1 regulated dc output voltage

```
//chapter26
//example26.1
//page570

R2=2.4d3 // ohm
R1=240 // ohm

V_out=1.25*(1+R2/R1)

printf("regulated dc output voltage = %.3f V \n",
   V_out)
```

Chapter 27

Hybrid parameters

Scilab code Exa 27.1 ratio

```
//chapter27
//example27.1
//page574

R1=10 // ohm
R2=5 // ohm

// for h11 and h21, imagine that output terminals are shorted hence it is clear that input impedence is equal to R1.
  // this is h11 by definition so
  h11=R1

  // now current will flow of same magnitude but in opposite directions through input and output terminals so output_current/input_current = -1
  // but this ratio is h21 by definition. Thus
  h21=-1

// for h12 and h22 imagine a voltage source on
```

```
    output terminals
    // this voltage will be avilable on input
       terminals also since current through 10 ohm
       resistor = 0.
    // hence input_voltage/output_voltage = 1
    // but this ratio is h12 by definition. Thus
    h12=1

    // here output impedence looking into output
       terminals with input terminals open is 5 ohm.
    // its reciprocal is h22 by definition. Thus
    h22=1/5

printf("h11 = %.3f ohm \n",h11)
printf("h21 = %.3f \n",h21)
printf("h12 = %.3f \n",h12)
printf("h22 = %.3f ohm \n",h22)
```

Scilab code Exa 27.2 impedence

```
//chapter27
//example27.2
//page575

R1=4 // ohm
R2=4 // ohm
R3=4 // ohm

// for h11 and h21, imagine that output terminals
   are shorted hence it is clear that input
   impedence is equal to R1+R2*R3/(R2+R3)
    // this is h11 by definition so
    h11=R1+R2*R3/(R2+R3)

    // now current will divide equally at junction
```

```
        of 4 ohm resistors so output_current/
        input_current = -0.5
    // but this ratio is h21 by definition. Thus
    h21=-0.5

// for h12 and h22 imagine a voltage source on
    output terminals
    // this voltage will be divided by a factor 2
    // hence input_voltage/output_voltage = 0.5
    // but this ratio is h12 by definition. Thus
    h12=0.5

    // here output impedence looking into output
        terminals with input terminals open is 8 ohm.
    // its reciprocal is h22 by definition. Thus
    h22=1/8

printf("h11 = %.3f ohm \n",h11)
printf("h21 = %.3f \n",h21)
printf("h12 = %.3f \n",h12)
printf("h22 = %.3f ohm \n",h22)
```

Scilab code Exa 27.3 voltage gain of circuit

```
//chapter27
//example27.3
//page578

h11=10
h12=1
h21=-1
h22=0.2
rL=5 // ohm

Zin=h11-(h12*h21/(h22+1/rL))
```

```
Av=-h21/(Zin*(h22+1/rL))

printf("input impedence = %.3f ohm \n",Zin)
printf("voltage gain of circuit = %.3f \n",Av)
```

Scilab code Exa 27.4 approximate current gain

```
//chapter27
//example27.4
//page581

hie=2000 // ohm
hoe=1d-4 // mho
hre=1d-3
hfe=50
rL=600 // ohm

Zin=hie-hre*hfe/(hoe+1/rL)
// here second term can be neglected compared to hie
    so
Zin_approx=hie

Ai=hfe/(1+hoe*rL)
// if hoe*rL << 1 then
Ai_approx=hfe

Av=-hfe/(Zin*(hoe+1/rL))
// negative sign indicates phase shift between input
    and output

printf("input impedence = %.3f ohm \n",Zin)
printf("current gain = %.3f \n",Ai)
printf("voltage gain = %.3f. Here negative sign
   indicates phase shift between input and output.\n
    \n",Av)
```

```
printf("approximate input impedence = %.3f ohm \n",
   Zin_approx)
printf("approximate current gain = %.3f \n",
   Ai_approx)
```

Scilab code Exa 27.5 input impedence

```
//chapter27
//example27.5
//page582

hie=1700 // ohm
hre=1.3d-4
hoe=6d-6 // mho
hfe=38
rL=2000 // ohm

Zin=hie-hre*hfe/(hoe+1/rL)

Ai=hfe/(1+hoe*rL)

Av=-hfe/(Zin*(hoe+1/rL))

printf("input impedence = %.3f ohm \n",Zin)
printf("current gain = %.3f \n",Ai)
printf("voltage gain = %.3f \n",-Av) // considering
   magnitude of Av,we neglect its negative sign and
   so we display -Av instead of Av
```

Scilab code Exa 27.6 voltage gain

```
//chapter27
```

```
//example27.6
//page582

hie=1500 // ohm
hre=4d-4
hoe=5d-5 // mho
hfe=50
Rc=10d3 // ohm
R_L=30d3 // ohm
R1=80d3 // ohm
R2=40d3 // ohm

rL=Rc*R_L/(Rc+R_L)
Zin=hie-hre*hfe/(hoe+1/rL)
Zin_stage=Zin*(R1*R2/(R1+R2))/(Zin+(R1*R2/(R1+R2)))

Av=-hfe/(Zin*(hoe+1/rL))

printf("input impedence = %.3f ohm \n",Zin_stage)
printf("voltage gain = %.3f \n",Av)

// the accurate answers are input impedence =
   1321.957 ohm and voltage gain = -196.078 but in
   book they are given as 1320 ohm and -196
   respectively
```

Scilab code Exa 27.7 micro mho

```
//chapter27
//example27.7
//page584

Vbe=10d-3 // V
Vbe2=0.65d-3 // V
Vce=1 // V
```

```
Ib=10d-6 // A
Ic=1d-3 // A
Ic2=60d-6 // A

hie=Vbe/Ib // in ohm
hfe=Ic/Ib // in ohm
hre=Vbe2/Vce
hoe=Ic2/Vce // in mho

printf("hie = %.3f ohm \n",hie)
printf("hfe = %.3f ohm \n",hfe)
printf("hre = %.5f \n",hre)
printf("hoe = %.3f micro mho \n",hoe*1d6)
```

Chapter 28

Digital electronics

Scilab code Exa 28.1 binary equivalent of decimal number 37

```
//chapter28
//example28.1
//page590

a= dec2bin (37)
disp(a,'binary equivalent of decimal number 37 = ')
```

Scilab code Exa 28.2 binary equivalent of decimal number 23

```
//chapter28
//example28.2
//page590

a= dec2bin (23)
disp(a,'binary equivalent of decimal number 23 = ')
```

Scilab code Exa 28.3 equivalent decimal of binary 110001

```
//chapter28
//example28.3
//page591

a= bin2dec ( ' 110001 ' )
printf("equivalent decimal of binary 110001 is %d \n
   ",a)
```

Scilab code Exa 28.4 page598

```
//chapter28
//example28.4
//page598

disp(" A        B        Y_dash = A + B      Y = Y_dash.A
     ")
disp(" 0        0              0                   0       ")
disp(" 1        0              1                   1       ")
disp(" 0        1              1                   0       ")
disp(" 1        1              1                   1       ")

printf("\nexplanation: \n")
printf("A=0 and B=0 give A'=1 and B'=1 so Y_dash = A
    + B is 0 and Y = Y_dash.A is 0 \n")

printf("A=1 and B=0 give A'=0 and B'=1 so Y_dash = A
    + B is 1 and Y = Y_dash.A is 1 \n")

printf("A=0 and B=1 give A'=1 and B'=0 so Y_dash = A
    + B is 1 and Y = Y_dash.A is 0 \n")

printf("A=1 and B=1 give A'=0 and B'=0 so Y_dash = A
    + B is 1 and Y = Y_dash.A is 1 \n")
```

Scilab code Exa 28.5 page598

```
//chapter28
//example28.5
//page598

disp(" A        B        A'      Y_dash = A'. B        B'
       Y = Y_dash + B' ")
disp(" 0        0        1         0                   1
         1 ")
disp(" 1        0        0         0                   1
         1 ")
disp(" 0        1        1         1                   0
         1 ")
disp(" 1        1        0         0                   0
         0 ")

printf("\nexplanation: \n")
printf("A=0 and B=0 give A'=1 and B'=1 so Y_dash = A
   '.B is 0 and Y = Y_dash + B' is 1 \n")

printf("A=1 and B=0 give A'=0 and B'=1 so Y_dash = A
   '.B is 0 and Y = Y_dash + B' is 1 \n")

printf("A=0 and B=1 give A'=1 and B'=0 so Y_dash = A
   '.B is 1 and Y = Y_dash + B' is 1 \n")

printf("A=1 and B=1 give A'=0 and B'=0 so Y_dash = A
   '.B is 0 and Y = Y_dash + B' is 0 \n")
```

Scilab code Exa 28.6 factorize

```
// chapter28
// example28.6
//page606

printf("Y = A . B . C` . D` + A` . B . C` . D` + A` . B . C . D` + A . B . C . D` \n")
printf("taking out the common factors \n")
printf("Y = B . C` . D` . ( A + A` ) + B . C . D` . ( A + A`) \n")
printf("By theorem 3 \n")
printf("Y = B . C` . D` + B . C . D` \n")
printf("again factorize \n")
printf("Y = B . D` ( C + C` ) \n")
printf("By theorem 3 \n")
printf("Y = B . D` . 1 \n")
printf("thus \n")
printf("Y = B . D` \n")
```

Scilab code Exa 28.7 Apply cummulative law and theorem

```
// chapter28
// example28.7
//page606

printf("Y = A . B + A . ( B + C ) + B . ( B + C ) \n")
printf("By thoerem 14 \n")
printf("Y = A . B + A . B + A . C + B . B + B .C \n")
printf("By theorem 6 \n")
printf("Y= A . B + A . B + A . C + B + B .C \n")
printf("By theorem 5 \n")
printf("Y = A . B + A . C + B + B . C \n")
printf("Factor B out of last 2 terms \n")
printf("Y = A . B + A . C + B . ( 1 + C ) \n")
```

```
printf("Apply cummulative law and theorem 7 \n")
printf("Y = A . B + A . C + B . 1 \n")
printf("Apply theorem 2 \n")
printf("Y = A . B + A . C + B \n")
printf("Factor B out of first and third terms \n")
printf("Y = B . ( A + 1 ) + A . C \n")
printf("Apply theorem 7 \n")
printf("Y = B . 1 + A . C \n")
printf("Apply theorem 2 \n")
printf("Y = B + A . C \n")
```

Scilab code Exa 28.8 Based on theorem 16 & 3

```
//chapter28
//example28.8
//page607

printf("i) Y = A + A' . B \n")
printf("   By theorem 16 \n")
printf("   Y = A + A . B + A' . B \n")
printf("     = A + B ( A + A') \n")
printf("   By theorem 3 \n")
printf("   Y = A + B \n \n")

printf("ii) Y = A . B + A' . C + B . C \n")
printf("      = A . B + A' . C + B . C ( A + A' )\n")
printf("      = A . B + A' . C + A . B . C + A' . B . C \n")
printf("      = A . B ( 1 + C ) + A' . C( 1 + B ) \n")
printf("      = A . B + A' . C \n")
```

Scilab code Exa 28.9 Using De Morgan theore

```
//chapter28
//example28.9
//page607

printf("Y = ( ( A + B )' . C . D' )' \n")
printf("Using De Morgan theorem \n")
printf("Y = ( A + B ) + C' + D \n")
printf("Y = A + B + C' + D \n")
```

Scilab code Exa 28.10 De Morgan theorem

```
//chapter28
//example28.10
//page607

printf("1) Y  = A . B . C' + A . ( B . C )' \n")
printf("   Y' = ( A . B . C' + A . ( B . C )' )' \n")
printf("   By De Morgan theorem \n")
printf("   Y' = ( A . B . C')' . ( A . ( B . C)' )' \n")
printf("   By De Morgan theorem \n")
printf("   Y' = ( A' + B' + C ) . ( A' + B + C ) \n \n")
printf("2) Y  = A' . ( B .C' + B' . C ) \n")
printf("   Y'  = ( A' . ( B .C' + B' . C ) )' \n")
printf("   By De Morgan theorem \n")
printf("   Y' = A + ( B . C' + B' .C )' \n")
printf("   By De Morgan theorem \n")
printf("   Y' = A + ( B . C')' . ( B' . C )' \n")
printf("   By De Morgan theorem \n")
printf("   Y' = A + ( B' + C ) . ( B + C') \n")
printf("   Y' = A + ( B . C )' + ( B . C ) \n")
```

Scilab code Exa 28.11 De Morgan theorem

```
//chapter28
//example28.11
//page608

printf("1) Y = ( A + B + C ) . ( A + B ) \n")
printf("    Y = A . A + A . B + B . A + B . B + C . A
    + C . B \n")
printf("    Using A . A = A we get \n")
printf("    Y = A + A . B + A . B + B + A . C + B . C
    \n")
printf("    Using A . B + A . B = A . B we get \n")
printf("    Y = A + A . B + B + A . C + B . C \n")
printf("    Using A + A . B = A we get \n")
printf("    Y = A + B + A . C + B . C \n")
printf("      = A . ( 1 + C ) + B . ( 1 + C ) \n")
printf("    Using 1 + C = 1 we get \n")
printf("    Y = A . 1 + B . 1 \n")
printf("    Y = A + B \n \n")

printf("2) Y = A . B + A . B . C + A . B . C` \n")
printf("      = A . B + A . B ( C + C` ) \n")
printf("    Since C + C` = 1 we get \n")
printf("    Y = A . B + A . B \n")
printf("      = A . B \n \n")


printf("3) Y = 1 + A . ( B . C` + B . C + B` . C`) +
    A . B` . C + A . C \n")
printf("    Using 1 + A = 1 and 1 + A . ( B . C` + B
    . C + ( B . C )` ) = 1 we get \n")
printf("    Y = 1 + A . B` . C + A . C \n")
printf("    Y = 1 + A . C \n")
```

```
printf("    Y = 1 \n \n")

printf("4) Y = ( ( A + B‘ + C ) + ( B + C‘ ))‘ \n")
printf("    By De Morgan theorem \n")
printf("    Y = ( A + B‘ + C )‘ . ( B + C‘ )‘ \n")
printf("    By De Morgan theorem \n")
printf("    Y = ( A‘ . B . C‘ ) . ( B‘ . C ) \n")
printf("    Since B . B‘ = 0 and C . C‘ = 0 we get \n
  ")
printf("    Y = 0 \n")
```

Scilab code Exa 28.12 theorem

```
//chapter28
//example28.12
//page609

printf(" Y = A . B‘ . D + A . B‘ . D‘ \n")
printf(" Factor out A . B‘ by theorem 14 \n")
printf(" Y = A . B‘ ( D + D‘ ) \n")
printf(" But by theorem 3 D + D‘ = 1 \n")
printf(" Y = A . B‘ . 1 \n")
printf(" By theorem 2 \n")
printf(" Y = A . B‘ \n")
```

Scilab code Exa 28.13 theorem

```
//chapter28
//example28.13
//page609

printf(" Y = ( A‘ + B ) . ( A + B ) \n")
printf(" By theorem 15 \n")
```

```
printf(" Y = A` . A + A` . B + B . A + B . B \n")
printf(" By theorem 4 and 6 \n")
printf(" Y = 0 + A` . B + B . A + B \n")
printf(" Y = A` . B + B . A + B \n")

printf(" By theorem 14 \n")
printf(" Y = B . ( A` + A + 1 ) \n")
printf(" By theorem 7 \n")
printf(" Y = B . ( A` + 1 ) \n")
printf(" By theorem 7 \n")
printf(" Y = B . 1 ) \n")
printf(" By theorem 2 \n")
printf(" Y = B \n")
```