```
Private Sub Frame2_Click()

End Sub

Private Sub TextBox10_Change()

End Sub

Private Sub TextBox14_Change()

End Sub

Private Sub TextBox17_Change()

End Sub

Private Sub TextBox2_Change()

End Sub

Private Sub TextBox20_Change()

End Sub

Private Sub TextBox21_Change()

End Sub

Private Sub TextBox22_Change()

End Sub

Private Sub TextBox23_Change()

End Sub

Private Sub TextBox24_Change()

End Sub

Private Sub TextBox25_Change()

End Sub

Private Sub TextBox26_Change()

End Sub

Private Sub TextBox27_Change()

End Sub

Private Sub TextBox28_Change()

End Sub

Private Sub TextBox29_Change()

End Sub

Private Sub TextBox3_Change()

End Sub

Private Sub TextBox30_Change()

End Sub

Private Sub TextBox31_Change()

End Sub

Private Sub TextBox32_Change()
```

```
End Sub

Private Sub TextBox33_Change()

End Sub

Private Sub TextBox34_Change()

End Sub

Private Sub TextBox35_Change()

End Sub

Private Sub TextBox36_Change()

End Sub

Private Sub TextBox37_Change()

End Sub

Private Sub TextBox38_Change()

End Sub

Private Sub TextBox4_Change()

End Sub

Private Sub TextBox40_Change()

End Sub

Private Sub TextBox5_Change()

End Sub

Private Sub TextBox6_Change()

End Sub

Private Sub TextBox7_Change()

End Sub

Private Sub TextBox8_Change()

End Sub

Private Sub TextBox9_AfterUpdate()

End Sub

Private Sub TextBox9_Change()

End Sub

Private Sub UserForm_Click()

End Sub

Private Sub UserForm_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

End Sub

Private Sub UserForm_Deactivate()

End Sub

Private Sub UserForm_Initialize()
```

```
End Sub

Private Sub UserForm_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)

End Sub

Private Sub UserForm_KeyUp(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)

End Sub

Private Sub UserForm_Layout()

End Sub

Private Sub UserForm_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal x As Single, ByV
al Y As Single)

End Sub

Private Sub UserForm_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal x As Single, ByV
al Y As Single)

End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)

End Sub

Private Sub UserForm_RemoveControl(ByVal Control As MSForms.Control)

End Sub

Private Sub UserForm_Resize()

End Sub



End Sub
Control Register for SCADA Switch
Dim ControlRegister As Integer
Dim PortStatus As Boolean

Sub ReadPort()
    ControlRegister = &H1A ' Example register address
    PortStatus = (ControlRegister And &H1) = &H1
    If PortStatus Then
        MsgBox "Port Active"
    Else
        MsgBox "Port Inactive"
    End If
End Sub
' LED connected to amplifier logic
Sub ControlLED(ByVal AmpLevel As Integer)
    If AmpLevel > 5 Then
        LEDPin = True
    Else
        LEDPin = False
    End If
End Sub
' Relay logic for fault detection
Dim FaultA, FaultB, FaultC As Boolean

Sub CheckFaults()
    If FaultA Or FaultB Or FaultC Then
        MsgBox "Fault Detected"
        ActivateBreaker()
    End If
End Sub

Sub ActivateBreaker()
    ' Simulate breaker trip
    BreakerStatus = "Tripped"
```

```
End Sub
[Start]
   Print
[Read Sensor Data]
   Print
[Check Threshold]
   ???(Yes)??> [Activate Output Pin 7]
   ???(No)???> [Log Data]
   Print
End
[Initialize System]
   Print
[Monitor Current Z]
   Print
[Detect Fault A/B/C]
   ???(Fault A)??> [Trip Breaker A]
   ???(Fault B)??> [Trip Breaker B]
   ???(Fault C)??> [Trip Breaker C]
   Print
[Log Fault Event]
   Print
End
Dim ControlRegister As Byte
Dim PortInput As Boolean

Sub ReadControlPort()
    ControlRegister = &H1A ' Example address
    PortInput = (ControlRegister And &H1) = &H1
    If PortInput Then
        MsgBox "Port Active"
    Else
        MsgBox "Port Inactive"
    End If
End Sub
?? 1B: SCADA Switch Control
Dim SCADASwitch As Boolean

Sub ToggleSCADASwitch()
    SCADASwitch = Not SCADASwitch
    If SCADASwitch Then
        MsgBox "SCADA Switch ON"
    Else
        MsgBox "SCADA Switch OFF"
    End If
End Sub
?? Visual Basic Processor & Relay Logic (Core Code)
?? 1A: Control Register - Lecture Port
Dim ControlRegister As Byte
Dim PortInput As Boolean


    ControlRegister = &H1A ' Example address
    PortInput = (ControlRegister And &H1) = &H1
    If PortInput Then
        MsgBox "Port Active"
    Else
        MsgBox "Port Inactive"
    End If
End Sub
?? 1B: SCADA Switch Control
Dim SCADASwitch As Boolean


    SCADASwitch = Not SCADASwitch
    If SCADASwitch Then
        MsgBox "SCADA Switch ON"
    Else
        MsgBox "SCADA Switch OFF"
    End If
End Sub
?? LED Control - Connect to Amp
vbnet
Dim LEDState As Boolean
Dim AmpLevel As Integer


    If AmpLevel > 5 Then
        LEDState = True
        MsgBox "LED ON"
    Else
        LEDState = False
        MsgBox "LED OFF"
    End If
End Sub
```

?? Output Switch - Pin 7 Logic

vbnet

```
Dim OutputPin7 As Boolean

Sub SwitchOffPin7()
    OutputPin7 = False
    MsgBox "Pin 7 Output OFF"
End Sub
```

?? Relay Current Z Logic (Z = rg + jxd)

vbnet

```
Structure Impedance
    Dim rg As Double
    Dim jxd As Double
End Structure


    Return New Complex(r, x)
End Function
```

?? Logigramme: Fault Detection & Relay Trigger

plaintext

```
[Start]
   Print
[Read Current Z]
   Print
[Compare Threshold]
   ???(Above Limit)??> [Trigger Relay]
   ???(Normal)???????> [Continue Monitoring]
   Print
[Log Event]
   Print
End
```

?? Algorigramme: SCADA Switch + LED + Output Pin

plaintext

```
[Initialize System]
   Print
[Check SCADA Switch]
   Print
[If ON]
   ???> [Read Amp Level]
   ???> [If Amp > 5 ? LED ON]
   ???> [Else ? LED OFF]
   Print
[Switch OFF Pin 7]
   Print
End
```

?? Integration with Curriculum Text Boxes

Text Box	Functionality	Visual Basic Logic
1A	Control Register	ReadControlPort()
1B	SCADA Switch	ToggleSCADASwitch()
LED/Amp	LED Control	ControlLED()
Pin 7	Output Switch	SwitchOffPin7()
Relay Z	Impedance Logic	CalculateZ()

?? Next Steps

Electrical Equations

"   Impedance: Z=rg+jxdZ = r_g + jx_d

"   Power flow: S=V?I=P+jQS = V \cdot I = P + jQ

"   Energy saving: E=i?t1t2E = i \int_{t_1}^{t_2}

"   Maximum power transfer: Pmax=V24ZP_{max} = \frac{V^2}{4Z}

"   Signal control: x?(t)=Ax(t)+Bu(t), Y(t)=Cx(t)+Du(t)\dot{x}(t) = Ax(t) + Bu(t),\ Y(t) = Cx(t) + Du(
t)

?? Visual Basic Integration

```
' Relay control based on impedance
```

```
Dim Z As Complex
Z = New Complex(rg, xd)

If Z.Magnitude > threshold Then
    ActivateRelay()
End If

Sub ActivateRelay()
    MsgBox "Relay Triggered"
End Sub
```

?? Logigramme & Algorigramme Mapping

?? Logigramme: Relay Activation

```
[Start]
   Print
[Measure Current Z]
   Print
[Compare with Threshold]
   ???(Above)??> [Trigger Relay]
   ???(Below)??> [Continue Monitoring]
   Print
End
```

?? Algorigramme: SCADA Switch + LED Control

plaintext

```
[Initialize]
   Print
[Check SCADA Switch]
   Print
[If ON]
   ???> [Read Amp Level]
   ???> [If Amp > 5 ? LED ON]
   ???> [Else ? LED OFF]
   Print
End
```

Modular Visual Basic Curriculum Framework (UserForm1)

?? Architecture Overview

"   UserForm1 hosts over 40+ TextBox controls.
"   Each TextBox maps to a curriculum module, technical function, or energy system descriptor.
"   Logic is grouped into KFrames (K1-K40), Trade Modules, Energy Systems, and SCADA Control.

?? TextBox Mapping Table

TextBox	Curriculum Domain	Technical Logic
TextBox1	SCADA Switch (1B)	ToggleSCADASwitch()
TextBox2	LED Control Description	ControlLED(AmpLevel)
TextBox3	VCC Level (5.0V)	If AmpLevel > 5 Then LED ON
TextBox4	Photovoltaic Installation	PV grid logic
TextBox5	General Protection (1D)	ActivateBreaker()
TextBox6	Data Acquisition (1E)	ReadSensorData()
TextBox7	Output Switch Pin 7	SwitchOffPin7()
TextBox20-29	Generator, Transformer, Motor, Capacitor Analysis (2A-2F)	CalculateZ(), EvaluateRelay()
TextBox30-38	Metering, Calibration, Performance, Stability (3A-4C)	P=V×I, S=P+jQ, Matrix(I1,I2)
TextBox40	Signal Processing, IoT, Energy Saving (4A-4B)	x?(t)=Ax(t)+Bu(t)
TextBox9	Revenue, Compliance, Planning	R = P > Q
TextBox10-19	Discovery Electronics, LED, Infrared, Alarm Timer, Flash, Sound, Detection	
TextBox22-24	PCB, UPS, Modicon, Eaton	Ampacity, Remote Generator
TextBox25-27	PV Grid, Cabling, Switchgear	Isolation, Overcurrent, Self Power
TextBox28-29	Career Workbook, Excel Tools	Load Switch, Component Sizing
TextBox31-33	Trade Theory, Syllabus, Compliance	Didactic Panel, Qualification
TextBox34-36	Career Discovery, Logic Control	Digital Libraries, Circuit Projects
TextBox37-38	Postgraduate Research, NRF, AIU Repository, Publication, Funding	

?? Sample Control Logic Snippets

SCADA Switch Control

```

    SCADASwitch = Not SCADASwitch
    MsgBox IIf(SCADASwitch, "SCADA Switch ON", "SCADA Switch OFF")
End Sub
```

LED Control

vb

```

    LEDState = (AmpLevel > 5)
    MsgBox IIf(LEDState, "LED ON", "LED OFF")
End Sub
```

Impedance Relay Logic

```
Type Impedance
```

```
    rg As Double
    jxd As Double
End Type


    CalculateZ = Sqr(Z.rg ^ 2 + Z.jxd ^ 2)
End Function
```

?? Curriculum Integration: KFrame Mapping

KFrame	Range	Domain
KFrame1	K1-K10	SCADA, LED, VCC, PV
KFrame2	K11-K20	Generator, Transformer, Faults
KFrame3	K21-K30	Metering, Performance, Stability
KFrame4	K31-K40	Signal Processing, IoT, Career

```
[Start]
   Print
[Read Sensor Data]
   Print
[Check Threshold]
   ??> (Yes) ? [Activate Output Pin 7]
   ??> (No) ? [Log Data]
   Print
End
```

Would you like me to scaffold this into a bilingual export sheet, GitLab-ready module, or printable dashboard for curriculum deployment? I can also help you generate logigramme/algorigramme diagrams or Visual Basic modules for each KFrame.

?? Modular SCADA Control Framework (UserForm1)

?? Core Variables

```
Dim ControlRegister As Byte
Dim PortStatus As Boolean
Dim SCADASwitch As Boolean
Dim LEDState As Boolean
Dim OutputPin7 As Boolean
Dim FaultA As Boolean, FaultB As Boolean, FaultC As Boolean
Dim BreakerStatus As String
Dim AmpLevel As Integer
```

?? Core Subroutines

?? Read Control Register (1A)

```
vb

    ControlRegister = &H1A
    PortStatus = (ControlRegister And &H1) = &H1
    MsgBox IIf(PortStatus, "Port Active", "Port Inactive")
End Sub
```

?? Toggle SCADA Switch (1B)

```
vb

    SCADASwitch = Not SCADASwitch
    MsgBox IIf(SCADASwitch, "SCADA Switch ON", "SCADA Switch OFF")
End Sub
```

?? LED Control Based on Amp Level

```

    LEDState = (AmpLevel > 5)
    MsgBox IIf(LEDState, "LED ON", "LED OFF")
End Sub
```

?? Output Pin 7 Logic

```

    OutputPin7 = False
    MsgBox "Pin 7 Output OFF"
End Sub
```

?? Fault Detection and Breaker Activation

```
vb

    If FaultA Or FaultB Or FaultC Then
        MsgBox "Fault Detected"
        ActivateBreaker
    End If
End Sub


    BreakerStatus = "Tripped"
    MsgBox "Breaker Tripped"
End Sub
```

?? Impedance Logic (Z = rg + jxd)

```
Type Impedance
    rg As Double
    jxd As Double
End Type


    CalculateZ = Sqr(Z.rg ^ 2 + Z.jxd ^ 2)
End Function


    If CalculateZ(Z) > threshold Then
        ActivateRelay
    End If
End Sub


    MsgBox "Relay Triggered"
End Sub
```

?? UserForm1 TextBox Mapping

TextBox	Functionality	Linked Logic
1A	Control Register	ReadControlPort()
1B	SCADA Switch	ToggleSCADASwitch()
LED	Amplifier LED Control	ControlLED(AmpLevel)
Pin 7	Output Switch	SwitchOffPin7()
Relay Z	Impedance Relay Logic	EvaluateRelay(Z, threshold)

```
[Start]
   Print
[Read Current Z]
   Print
[Compare Threshold]
   ??> (Above) ? [Trigger Relay]
   ??> (Below) ? [Continue Monitoring]
   Print
[Log Event]
   Print
End
```

?? Algorigramme: SCADA + LED + Pin 7

```
[Initialize System]
   Print
[Check SCADA Switch]
   Print
[If ON]
   ??> [Read Amp Level]
   ??> [If Amp > 5 ? LED ON]
   ??> [Else ? LED OFF]
   Print
[Switch OFF Pin 7]
   Print
End
```

?? Curriculum Integration (Tab1/Tab2 Mapping)

Tab	TextBox	Description
Tab1	1B	SCADA Switch Control
Tab1	LED	LED connected to amplifier
Tab1	VCC	Voltaic power level
Tab1	1D	General protection
Tab1	1E	Data acquisition
Tab1	Pin 7	Output switch logic
Tab2	2A	Generator, transformer, capacitor analysis
Tab2	2B	Fault A/B/C, breaker, cable
Tab2	2C	Power station, ZTHV/XTHV, UPS
Tab2	2D	Network analysis, instrumentation
Tab2	2E	Power transfer, delta systems
Tab2	2F	Transformer connection, line instrumentation

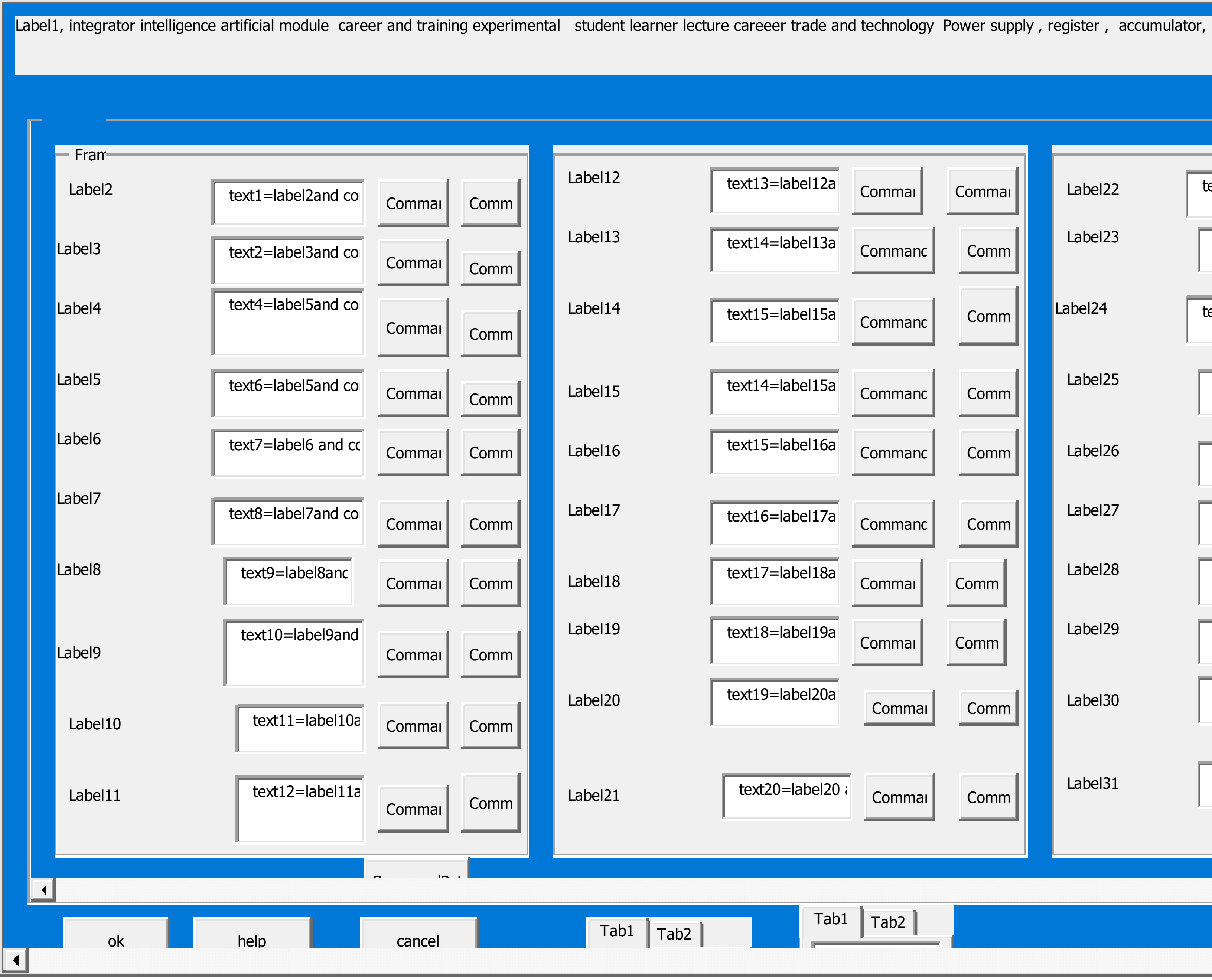
Would you like me to scaffold this into a bilingual print

?? Modular SCADA Control Framework (VBA)

?? Core Variables and Structures

```
Dim ControlRegister As Byte
Dim PortStatus As Boolean
Dim SCADASwitch As Boolean
Dim LEDState As Boolean
Dim OutputPin7 As Boolean
Dim FaultA As Boolean, FaultB As Boolean, FaultC As Boolean
Dim BreakerStatus As String
Dim AmpLevel As Integer

Type Impedance
    rg As Double
    jxd As Double
End Type

Dim Z As Impedance
?? Core Subroutines
1A: Read Control Register
vb
()
    ControlRegister = &H1A
    PortStatus = (ControlRegister And &H1) = &H1
    MsgBox IIf(PortStatus, "Port Active", "Port Inactive")
End Sub
1B: Toggle SCADA Switch

    SCADASwitch = Not SCADASwitch
    MsgBox IIf(SCADASwitch, "SCADA Switch ON", "SCADA Switch OFF")
End Sub
LED Control Based on Amplifier Level
vb

    LEDState = (AmpLevel > 5)
    MsgBox IIf(LEDState, "LED ON", "LED OFF")
End Sub
Output Pin 7 Logic

    OutputPin7 = False
    MsgBox "Pin 7 Output OFF"
End Sub
Fault Detection and Breaker Activation
vb

    If FaultA Or FaultB Or FaultC Then
        MsgBox "Fault Detected"
        ActivateBreaker
    End If
End Sub


    BreakerStatus = "Tripped"
    MsgBox "Breaker Tripped"
End Sub
Print Impedance - Based; Relay; Logic
vb

    CalculateZ = Sqr(rg ^ 2 + jxd ^ 2)
End Function


    If CalculateZ(Z.rg, Z.jxd) > threshold Then
        ActivateRelay
    End If
End Sub


    MsgBox "Relay Triggered"
End Sub
?? UserForm Integration (TextBox Mapping)
TextBox Functionality   Linked Subroutine
1A  Control Register    ReadControlPort()
1B  SCADA Switch    ToggleSCADASwitch()
LED Amplifier LED Control   ControlLED(AmpLevel)
Pin 7   Output Switch   SwitchOffPin7()
Relay Z Impedance Relay Logic   EvaluateRelay(threshold)
?? Logigramme: Fault Detection
[Start]
   Print
[Read Current Z]
   Print
[Compare Threshold]
```

```
   ??> (Above) ? [Trigger Relay]
   ??> (Below) ? [Continue Monitoring]
   Print
[Log Event]
   Print
End
```



Label1, integrator intelligence artificial module career and training experimental student learner lecture careeer trade and technology Power supply , register , accumulator,

text1=label2and co
text2=label3and co
text4=label5and co
text6=label5and co
text7=label6 and cc
text8=label7and co
text9=label8anc
text10=label9and
text11=label10a
text12=label11a
text13=label12a
text14=label13a
text15=label15a
text14=label15a
text15=label16a
text16=label17a
text17=label18a
text18=label19a
text19=label20a
text20=label20 a