Subnetting Tutorial

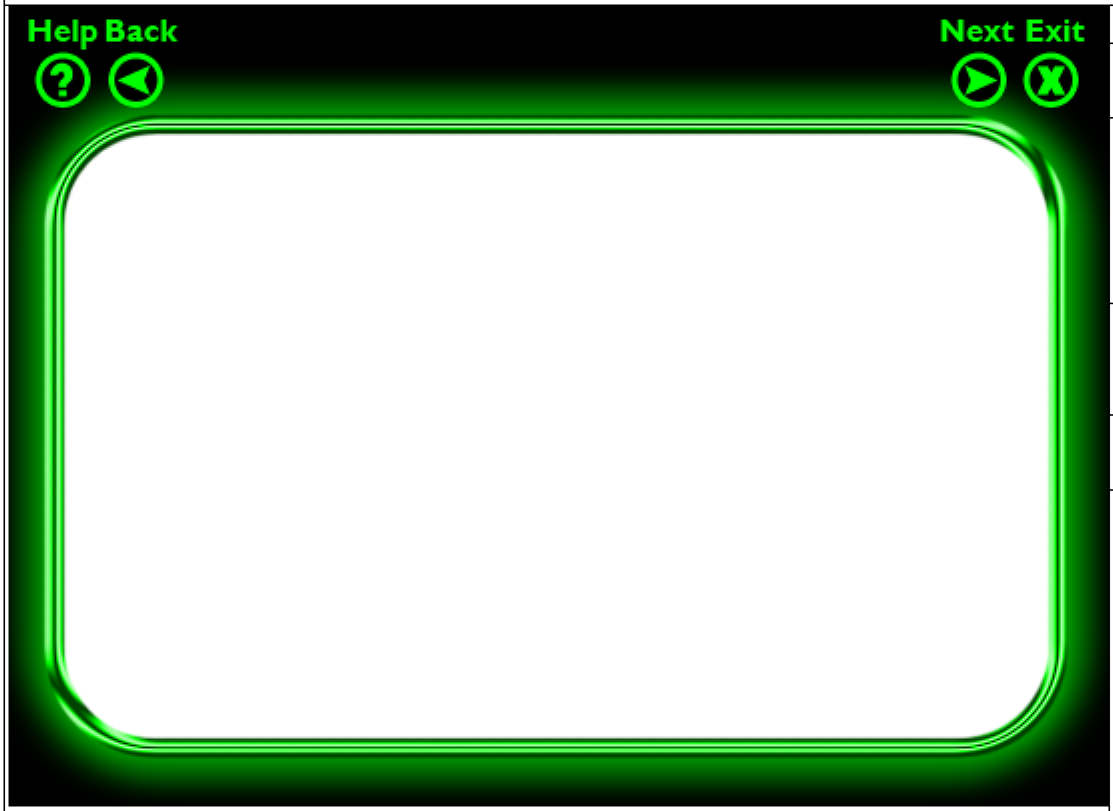
Project Prototype

IDT 520

California State University Fullerton

The target audience for the Subnetting Tutorial is any student who is taking part in a networking fundamentals course as subnetting is one of the basic skills that networking students need to learn. The objectives students will be able to achieve from this tutorial include being able to: evaluate network configurations, analyze subnet mask changes needed in order to apply subnetting to a network, and deduce the correct IP addresses to be applied to networked systems participating in a subnet. The user’s answers to each step of the subnetting process will be assessed programmatically each time they submit their answers for evaluation. By using Flash and its ActionScript programming capabilities the Subnetting Tutor will allow students to practice their subnetting skills at any time and they will be provided with automated feedback during their practice sessions.

Subnetting Tutorial Prototype **Screen 0** (bkg for all screens) **3/10/10**



Specifications:

Stage color: black; **size:** 550 x 400 px;

Button circle border & icon color: rgb(0,255,0). **Button circle fill color:** rgb(0,0,0); **Button circle border stroke:** 3px

Button names: btnHelp, btnBack, btnNext, btnExit

Rollover font: Gil Sans 12 pt. bold rgb(0,255,0)

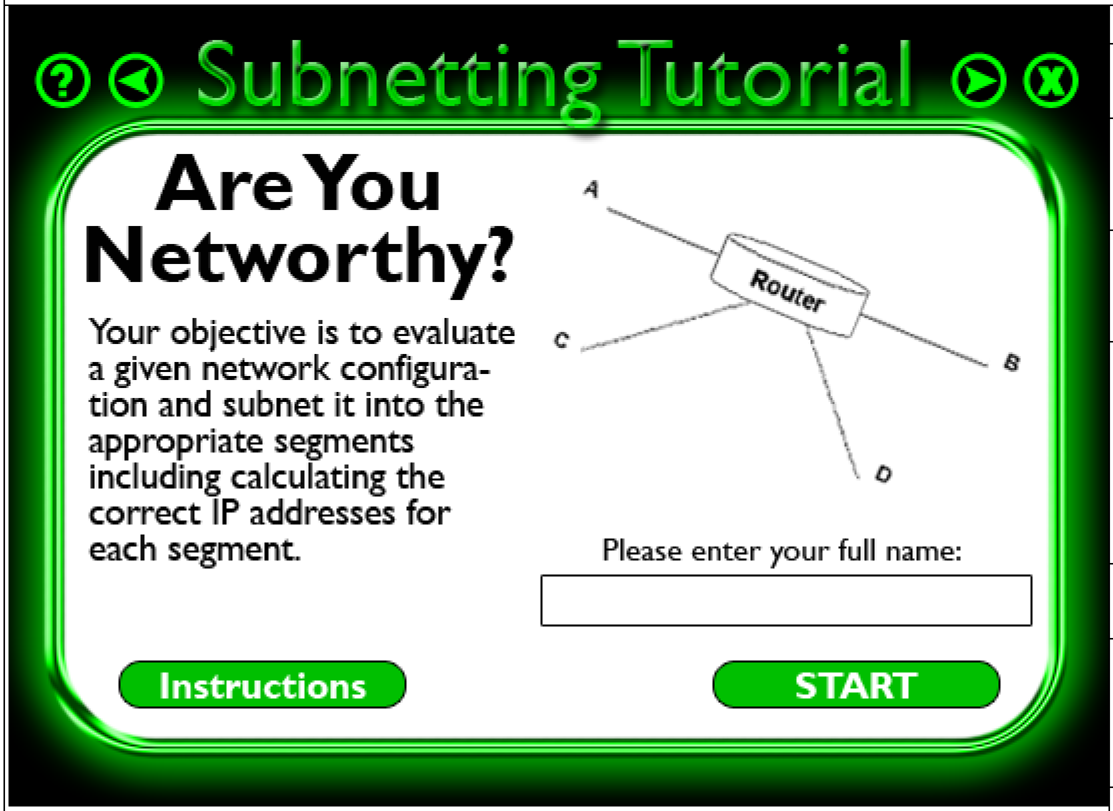
Content border: color rgb(0,255,0); width 500 px; height 320 px; rounded-corners 40; position X:25 Y:50; **Bevel:** blurX = 5px, blurY = 5px; strength = 100%, quality = high, shadow rgb(0,0,0), highlight rgb(255,255,255), angle 45°, distance 5px, knockout = unchecked, type = full; **Glow:** blurX = 30px, blurY = 30px, strength = 100%, quality = high; color = rgb(0,255,0), knockout = unchecked, inner glow = unchecked

Comments:

This is the background screen which all other screens will display on top of. There will be no menus per-se in this program. All user functions will be controlled by the Help, Back, Next, & Exit buttons at the top of the screen and the buttons(s) included in the content area of each screen. Each button will display the text shown when it's over state occurs. Button images are to be created using Photoshop symbols and imported into Flash as PNG files, their color will be rgb(0,255,0). Although the buttons will appear on most screens they will be programmatically removed from the display list on the Instruction, Error, and Help. Also, the Back button will not be displayed on the Title screen and the Next button will not be displayed on the Exit screen.

The title for each page (not shown here) will be based on the value of the variable: **imgPageTitle**. 5 PNG images will be used for page titles: SubnettingTutorial.png, Goodbye.png, Error.png, Help.png, & Instructions.png. Each active screen will be tracked by the **var screen:Number** which will be the two-digit value of the screen number.

Subnetting Tutorial Prototype Screen 1 3/10/10



Are You Networthy?

Your objective is to evaluate a given network configuration and subnet it into the appropriate segments including calculating the correct IP addresses for each segment.

Please enter your full name:

Buttons located in screen content area will all be created in Flash. Button width will be 150 pixels, height 20 pixels, fill color rgb(0,255,0), border is 1 pixel black, and text will be Gills Sans 14 bold white. During hover button fill color should turn black and text should turn rgb(0,255,0).

Specifications:
imgPageTitle = SubnettingTutorial.png;
Page Head text type: Static; **font:** Gills Sans 30 pt bold black
Body text type: Static; **font:** Gills Sans 14 pt regular black

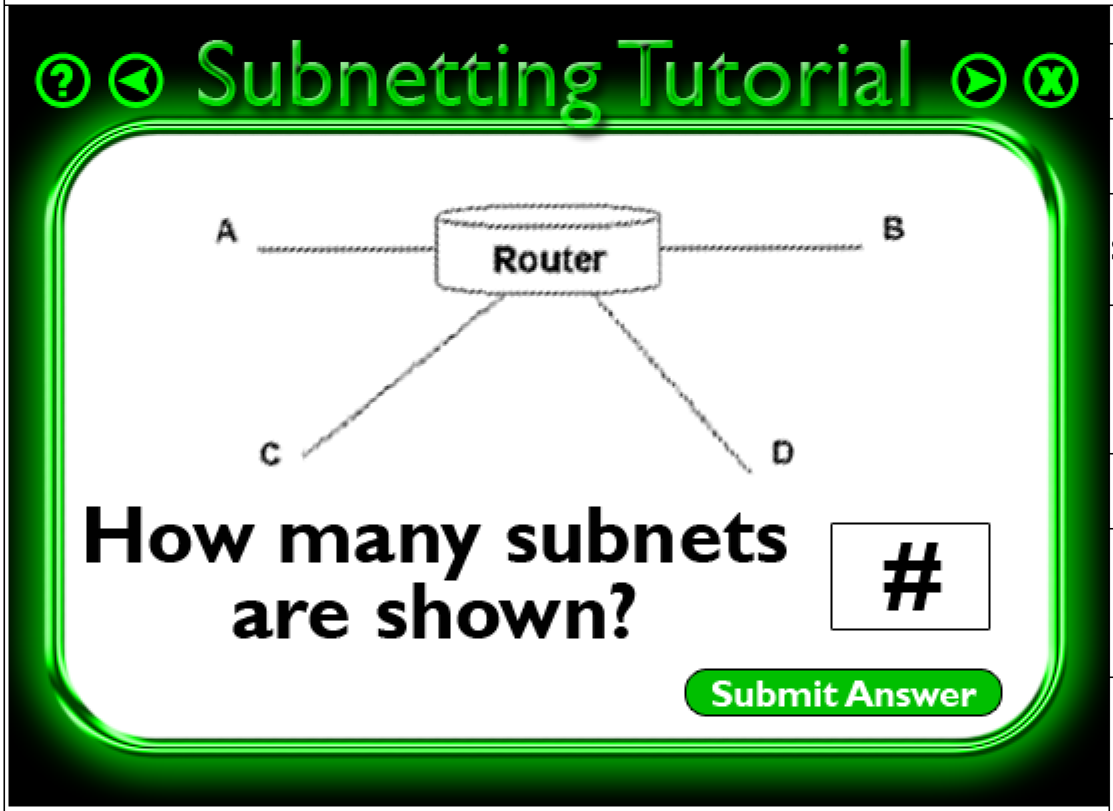
Textfield type: Input; **font:** Gills Sans 18 pt bold black;
name: txtUserName;
label = “Please enter your full name:”

Button name: btnInstructions
Event listener for btnInstructions: CLICK; **Calls:** gotoAndStop(Screen11)
Button name: btnStart
Event listeners for btnStart: CLICK & ENTER; **Calls:** AnswerChecker()
Function: AnswerChecker() evaluates if Textfield is empty. If **TRUE** gotoAndStop(screen09) pass var screen, If **FALSE** call GenerateSubnetConfig()
Function: GenerateSubnetConfig() Generate subnet configuration; **vars:** subnetImage, totalSubnets, subnetX, networkAddress, CIDR, maskBits, newMask, firstIP, lastIP, broadcastIP; gotoAndStop(Screen02)

Comments:

This is the title screen for the program; it also serves the role of providing the tutorial objectives and retrieving user identification. In prototype version **GenerateSubnetConfig()** will select from three pre-defined subnet configurations: 2 subnets, 3 subnets, or 4 subnets each starting with a network address/CIDR of 192.168.0.0/16.

Subnetting Tutorial Prototype Screen 2 3/10/10



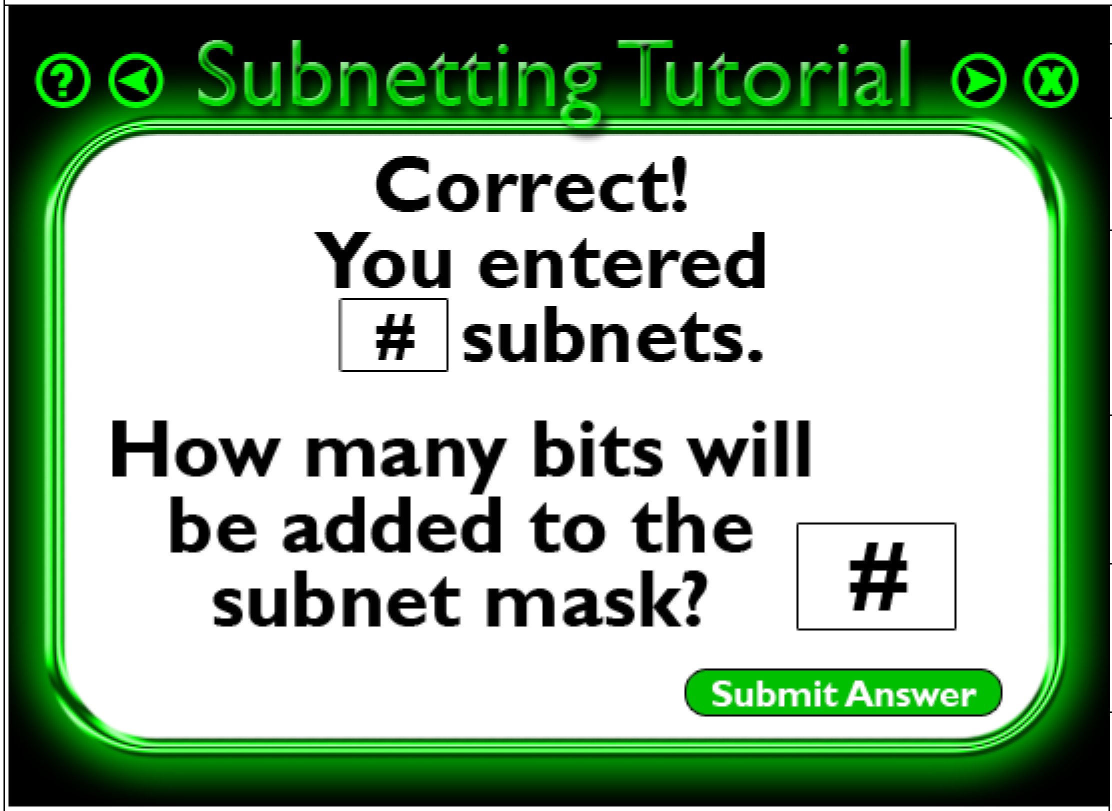
How many subnets are shown?

Specifications:
imgPageTitle = SubnettingTutorial.png;
Screen2Image = config var subnetImage
Page Head text type: Static; **font:** Gills Sans 30 pt bold
Textfield type: Input; **font:** Gills Sans 24 pt. bold; **name:** txtSubnetCount
Button name: btnSubmitAnswer
Event listeners for btnSubmitAnswer: CLICK & ENTER **Calls**: AnswerChecker()
Function: AnswerChecker() compare var userSubnets to config var totalSubnets; If **TRUE** gotoAndStop(Screen03), **FALSE** gotoAndStop(Screen09) pass var screen

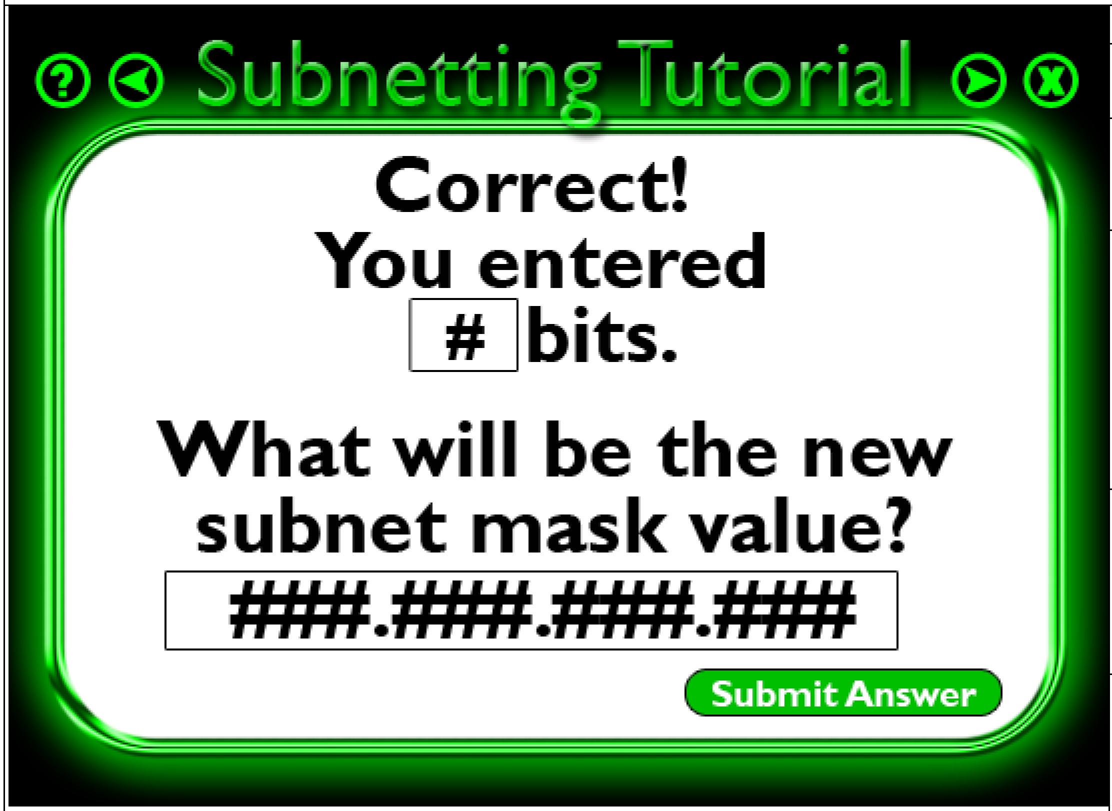
Comments:

In prototype subnet config choices will be 2, 3, or 4 subnets. 3 png files need to be created similar to example shown in this screen and will be assigned randomly to the var **imgRouterConfig:Loader**. As an alternative in the prototype 3 different Flash screens could be created, one for each subnet, and randomly selected at the beginning of each tutorial. **GenerateSubnetConfig() vars** would have to be assigned accordingly.

Subnetting Tutorial Prototype Screen 3 3/10/10

	Specifications:
 Correct! You entered # subnets. How many bits will be added to the subnet mask? #	**imgPageTitle** = SubnettingTutorial.png;
	Page Head text type: Static; **font:** Gills Sans 30 pt bold
	Textfield type: Dynamic **font:** Gills Sans 18 pt. bold; **name:** txtSubnetCountAnswer; **text** = userSubnets.text
	Textfield type: Input; **font:** Gills Sans 24 pt. bold; **name:** txtAddMaskBits
	Button name: btnSubmitAnswer (same object from screen 2)
	Event listeners for btnSubmitAnswer: CLICK & ENTER; **Calls:** AnswerChecker()
	Function: AnswerChecker() compare var userMaskBits to config var maskBits; If **TRUE** gotoAndStop(Screen04), **FALSE** gotoAndStop(Screen09) pass var screen
Comments:	

Subnetting Tutorial Prototype Screen 4 3/10/10



Correct!
You entered
# bits.

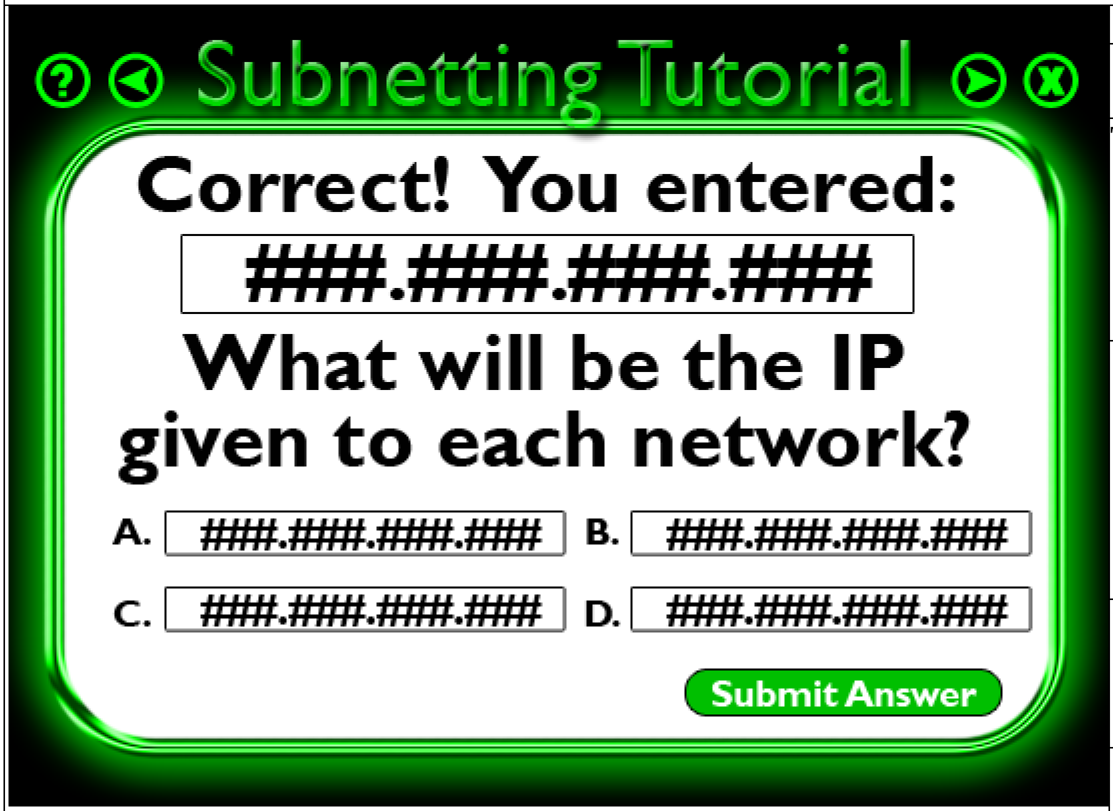
What will be the new subnet mask value?

###.###.###.###

Specifications:
imgPageTitle = SubnettingTutorial.png;
Page Head text type: Static; **font:** Gills Sans 30 pt bold
Textfield type: Dynamic **Textfield font:** Sans Serif 18 pt. bold **Textfield name:** txtAddMaskBitsAnswer **Textfield.text** = txtAddMaskBits.text
Textfield type: Input **Textfield text:** Sans Serif 24 pt. bold **Textfield name:** txtNewMask
Button name: btnSubmitAnswer (same object from screens 2 & 3)
Event listeners for btnSubmitAnswer: CLICK & ENTER **Calls:** AnswerChecker()
Function: AnswerChecker() compare var userNewMask to config var newMask; If **TRUE** gotoAndStop(Screen05), **FALSE** gotoAndStop(Screen09) pass var screen

Comments:

Subnetting Tutorial Prototype **Screen 5** **3/10/10**



Subnetting Tutorial

Correct! You entered:

###.###.###.###

What will be the IP given to each network?

A. ###.###.###.### B. ###.###.###.###

C. ###.###.###.### D. ###.###.###.###

Specifications:

imgPageTitle = SubnettingTutorial.png;

Textfield type: Dynamic; **font:** Sans Serif 24 pt. bold; **name:** txtNewMaskAnswer **text** = txtAddMaskBits.text

Textfield type: Input; **font:** Sans Serif 14 pt. bold; **names:**
txtNetworkA
txtNetworkB
txtNetworkC
txtNetworkD

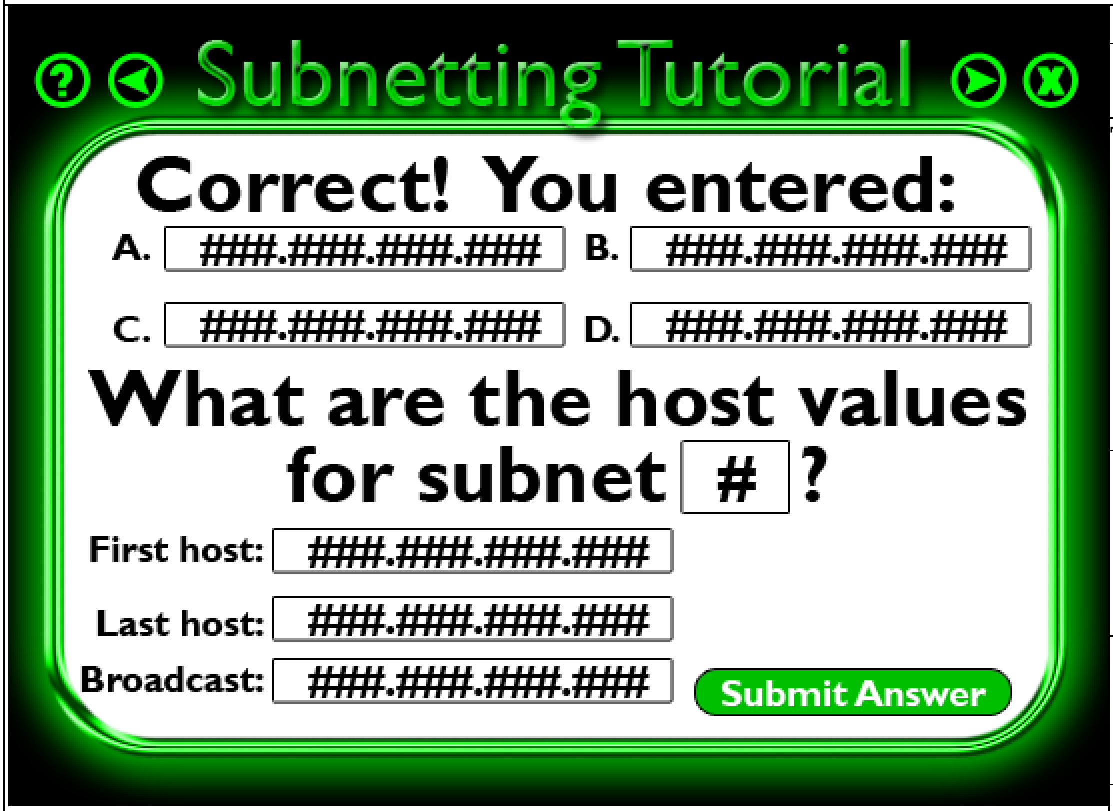
Button name: btnSubmitAnswer (same object from screens 2, 3 & 4)

Event listeners for btnSubmitAnswer: CLICK & ENTER **Calls:** AnswerChecker()

Function: AnswerChecker() compare var userSubnetA to config var subnetA (repeat for totalSubnets); If **TRUE** gotoAndStop(Screen06), **FALSE** gotoAndStop(Screen09) pass var screen

Comments:

Subnetting Tutorial Prototype Screen 6 3/10/10



Correct! You entered:

A. ###.###.###.### B. ###.###.###.###

C. ###.###.###.### D. ###.###.###.###

What are the host values for subnet # ?

First host: ###.###.###.###

Last host: ###.###.###.###

Broadcast: ###.###.###.###

Specifications:

imgPageTitle = SubnettingTutorial.png;

Textfield type: Dynamic; **font:** Sans Serif 18 pt. bold; **names:** txtNetworkAAnswer, txtNetworkBAnswer, txtNetworkCAnswer, txtNetworkDAnswer; **text** = txtAddMaskBits.text

Textfield type: Input; **font:** Sans Serif 14 pt. bold; **names:** txtFirstHost, txtLastHost, txtBroadcast

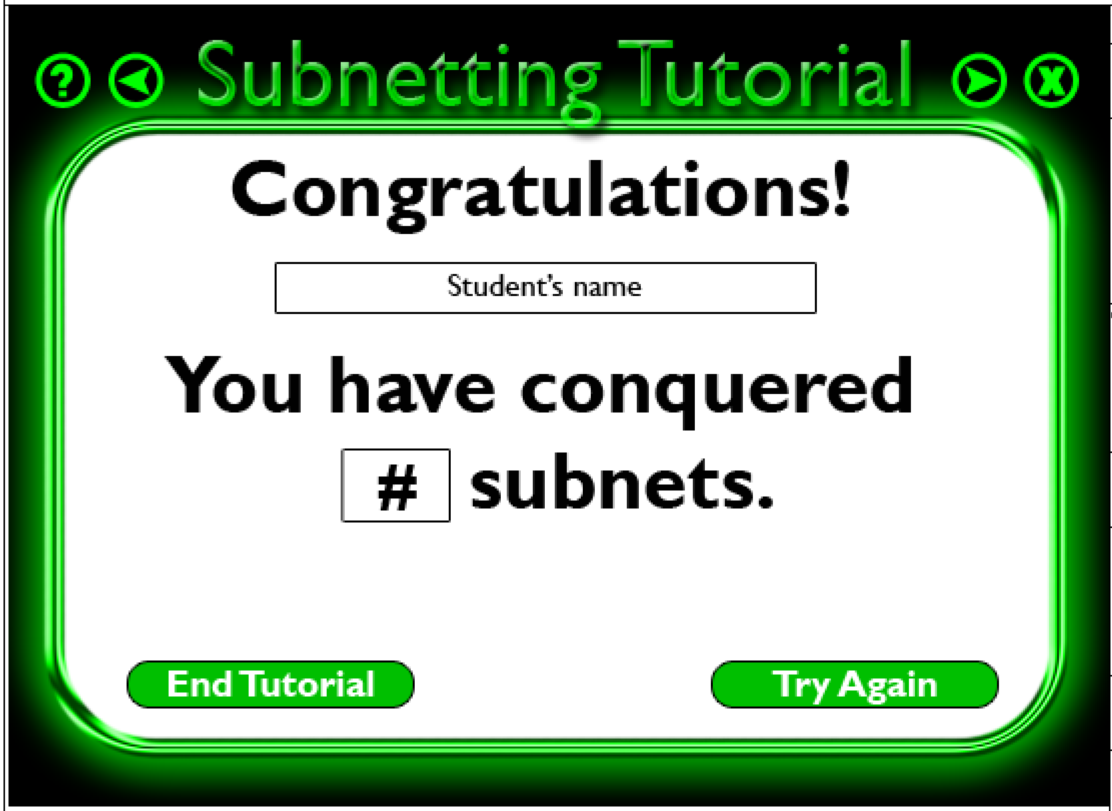
Button name: btnSubmitAnswer (same object from screens 2, 3, 4 & 5)

Event listeners for btnSubmitAnswer: CLICK & ENTER **Calls:** AnswerChecker()

Function: AnswerChecker() compare var userFirstIP, userLastIP, & userBroadcastIP to config vars firstIP, lastIP, broadcastIP; Repeat for totalSubnets; If all return **TRUE** gotoAndStop(Screen07), **ELSE** gotoAndStop(Screen09) pass var screen

Comments:

Subnetting Tutorial Prototype Screen 7 3/10/10



Congratulations!

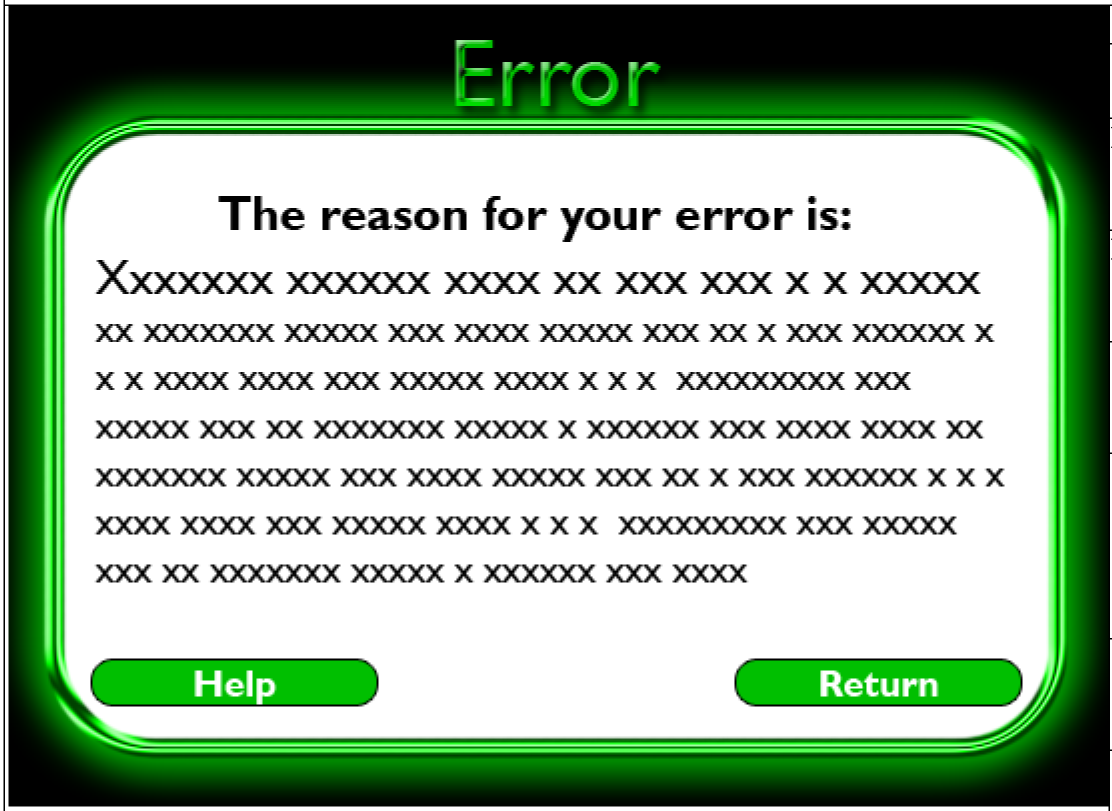
Student's name

You have conquered # subnets.

Specifications:
imgPageTitle = SubnettingTutorial.png;
Textfield type: Dynamic **font:** Sans Serif 18 pt bold; **name:** txtStudentName **text** = txtUserName.text
Textfield type: Dynamic; **font:** Sans Serif 14 pt. bold; **name:** totalSubnets (from config)
Button name: btnTryAgain
Event listeners for btnTryAgain: CLICK & ENTER **Calls:** AnswerChecker()
Button name: btnEndTutorial
Event listeners for btnEndTutorial: CLICK **Calls:** gotoAndStop(Screen08)

Comments:

Subnetting Tutorial Prototype Screen 9 3/10/10



The reason for your error is:

Xxxxxxx xxxxxx xxxx xx xxx xxx x x xxxxx xx xxxxxxx xxxxx xxx xxxx xxxxx xxx xx x xxx xxxxxx x x x xxxx xxxx xxx xxxxx xxxx x x x xxxxxxxxx xxx xxxxx xxx xx xxxxxxx xxxxx x xxxxxx xxx xxxx xxxx xx xxxxxxx xxxxx xxx xxxx xxxxx xxx xx x xxx xxxxxx x x x xxxx xxxx xxx xxxxx xxxx x x x xxxxxxxxx xxx xxxxx xxx xx xxxxxxx xxxxx x xxxxxx xxx xxxx

Specifications:

imgPageTitle = Error.png;

Heading text type: Static **font:** Gills Sans 18 pt. bold

Body text type: Dynamic font: Gills Sans 14 pt. regular

Button name: btnReturn **font:** Gills Sans 12 pt bold white

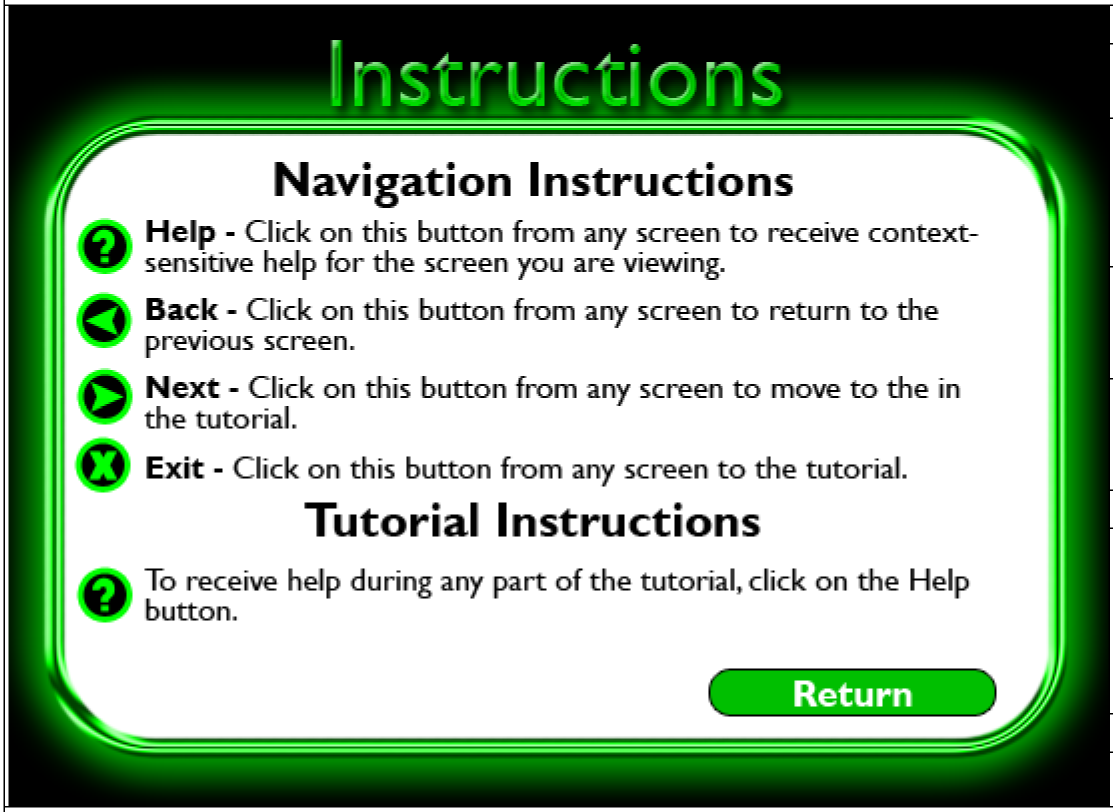
Event listener for btnReturn: CLICK & ENTER **Calls:** gotoAndPlay(screen)

Button name: btnHelp **font:** Gills Sans 12 pt bold white

Event listener for btnHelp: CLICK **Calls:** gotoAndPlay(Screen10) pass var screen

Comments:

Subnetting Tutorial Prototype Screen 11 3/10/10



Instructions

Navigation Instructions

Help - Click on this button from any screen to receive context-sensitive help for the screen you are viewing.

Back - Click on this button from any screen to return to the previous screen.

Next - Click on this button from any screen to move to the in the tutorial.

Exit - Click on this button from any screen to the tutorial.

Tutorial Instructions

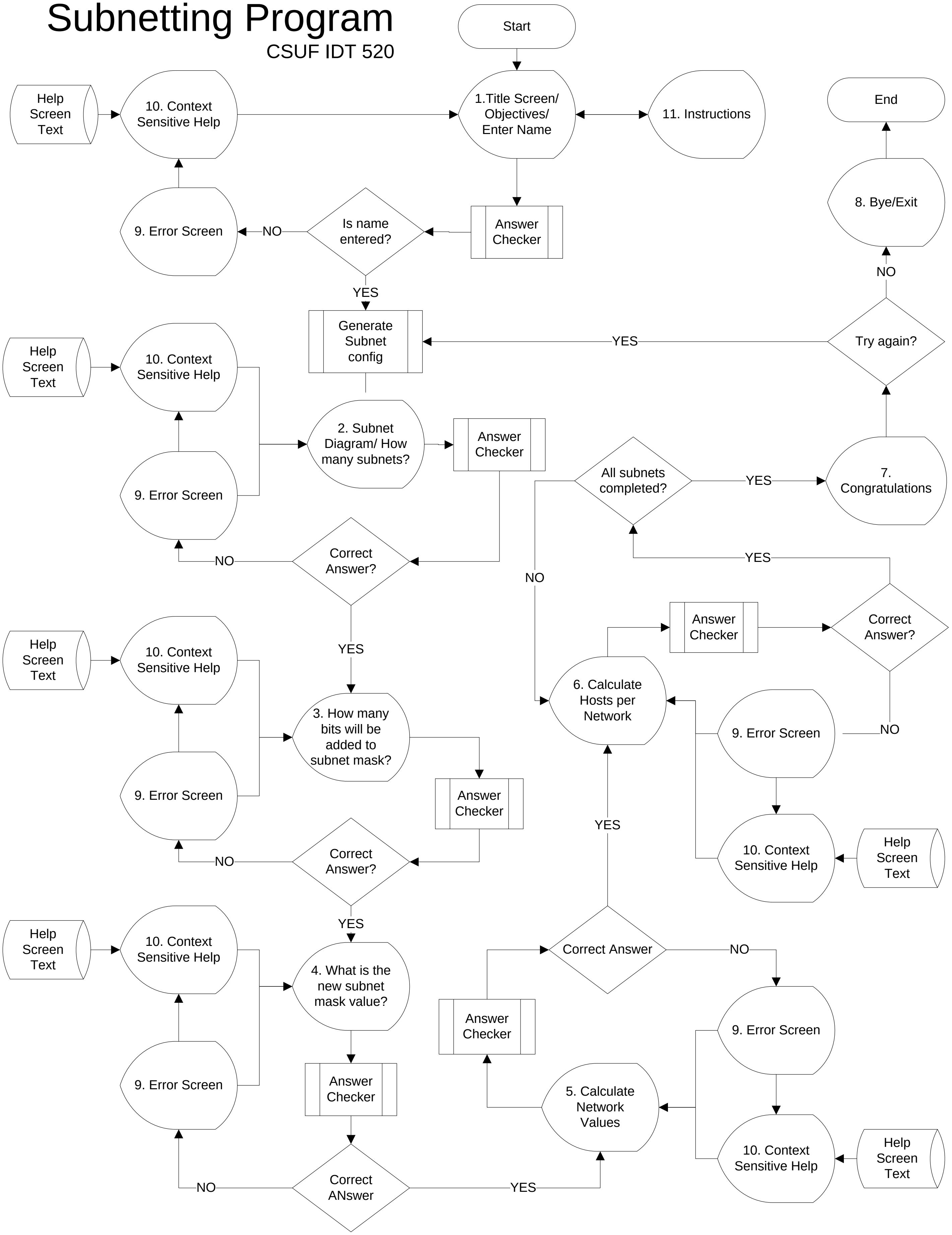
To receive help during any part of the tutorial, click on the Help button.

Specifications:
imgPageTitle = Instructions.png;
Section head text type: Static **font:** Gills Sans 18 pt bold
Button title text: Static **font:** Gills Sans 12 pt bold white
Instruction text: Static **font:** Gills Sans 12 pt regular
Button name: btnReturn
Event listener for btnReturn: CLICK & ENTER **Calls:** gotoAndPlay(Screen01)

Comments:

Subnetting Program

CSUF IDT 520


Start
1.Title Screen/ Objectives/ Enter Name
11. Instructions
Help Screen Text
10. Context Sensitive Help
9. Error Screen
NO
Is name entered?
Answer Checker
YES
Generate Subnet config
End
8. Bye/Exit
NO
Try again?
YES
Help Screen Text
10. Context Sensitive Help
9. Error Screen
2. Subnet Diagram/ How many subnets?
Answer Checker
All subnets completed?
YES
7. Congratulations
NO
Correct Answer?
NO
YES
YES
Answer Checker
Correct Answer?
Help Screen Text
10. Context Sensitive Help
9. Error Screen
3. How many bits will be added to subnet mask?
6. Calculate Hosts per Network
9. Error Screen
NO
Answer Checker
Correct Answer?
NO
YES
10. Context Sensitive Help
Help Screen Text
YES
Help Screen Text
10. Context Sensitive Help
9. Error Screen
4. What is the new subnet mask value?
Correct Answer
NO
Answer Checker
9. Error Screen
Answer Checker
5. Calculate Network Values
10. Context Sensitive Help
Help Screen Text
Correct ANswer
NO
YES