SY201 Name(s): _____

Alpha(s): _____

Assignment Type:	Lab	Collaboration Policy:	Default
Assignment Title:	Lab 3	Submit Project Name:	lab3
Electronic submission due: 2359 on 16 Sep Paper submission due: start of class 18 Sep Submission instructions: http://courses.cyber.usna.edu/SY201/calendar.php?load=policy			

1. Assignment Overview

Read through this entire assignment before starting work. In this assignment you will create a program that accepts a port number as user input and outputs information about the protocol that uses that port.

You will research basic information about the ports and associated protocols from IANA and Wikipedia. The Internet Assigned Numbers Authority (IANA) maintains the port number registration, among other Internet addressing data; IANA is the primary source. The Wikipedia article, merely displays the IANA maintained data in an easy to read and search format. Both web pages are linked to from the assignment web page.

2. Background Research

a. Read the introduction of the IANA Service Name and Transport Protocol Port Number Registry and complete the below table.

Note: System, user, and dynamic are the official names; well-known, registered, and ephemeral, respectively, are common names often used in place of the official names.

Table 1 - Port Number Ranges		
Range Name	Low Port Number	High Port Number
System		
User		
Dynamic		
Overall valid		

b. Search through the Wikipedia *List of TCP and UDP Port Numbers* and complete the following table.

For some protocols you may need to follow links to read more about the protocol.

You will likely need to follow the links from the Wikipedia page and read the descriptions of the protocols to determine whether the protocol is secure or not. Write True or False. Secure indicators include: (1) secure in the protocol name or (2) use of encryption. Insecure indicators include: (1) protocols that do not use encryption, (2) protocols used for games, or (3) when a separate port is used for secure version of protocol. Not all protocols that you think should be secure are secure.

The category assignments are specific to SY201, and generally categorize the protocol based on the primary intended use of the protocol:

- C*omms*: The protocol is generally used to provide communication between end users.
- *Data*: The protocol is generally used to provide data transfers between systems.
- *Game*: The protocol is generally used as part of an online game service.
- *Ops*: The protocol is generally used for information system configuration, coordination, or event logging.
- *Remote*: The protocol is generally used to provide remote access to a system.

Table 2 - Protocol Information

Port	Acronym / Short Name	Long (Full) Name	Category	Secure
20			Data	
22			Remote	
23	No Acronym		Remote	
53			Data	
67-68	DHCP		Ops	
80			Data	
123			Ops	
443			Data	
445	SMB		Data	True
514	No Acronym	Syslog	Ops	
666	No Acronym		Game	
6665-6669			Comms	

SY201 Name(s): _____

Alpha(s): _____

3. Specification - create a file lab3.py with the following functions and **includes comments adjacent to each function that describe the function's purpose and use**:

   a. Write a function named port that:
      i. Takes one argument: a valid port number (integer)
      ii. Returns the string "Port: [PORT_NUMBER]"

   b. Write a function named port_range that:
      i. Takes one argument: an integer
      ii. Returns the string "Range: [RANGE_NAME]" where RANGE_NAME is the range name associated with that port number from Table 1
      iii. If the port number is not included in Table 1, return the string 'Range: Unknown'

   c. Write a function named category that:
      i. Takes one argument: a valid port number (integer)
      ii. Returns the string "Category: [CATEGORY]" where CATEGORY is the category associated with that port number from Table 2
      iii. If the port number is not included in Table 2, return the string 'Category: Unknown'

   d. Write a function named protocol_short that:
      i. Takes one argument: a valid port number (integer)
      ii. Returns the string "[ACRONYM/SHORT_NAME]" where ACRONYM/SHORT_NAME is the acronym / short name associated with that port number from Table 2, or "No Acronym"
      iii. If the port number is not included in Table 2, return the string 'Unknown'

   e. Write a function named protocol_long that:
      i. Takes one argument: a valid port number (integer)
      ii. Returns the string "[LONG/FULL_NAME]" where LONG/FULL_NAME is the long / full name associated with that port number from Table 2
      iii. If the port number is not included in Table 2, return the string 'Unknown'

   f. Write a function named entire_protocol that:
      i. Takes one argument: a valid port number (integer)
      ii. Calls the functions protocol_short and protocol_long
      iii. Returns the string "Protocol: [ACRONYM/SHORT_NAME] - [LONG/FULL_NAME]" where ACRONYM/SHORT_NAME and LONG/FULL_NAME are the acronym / short name and long / full name, respectively, associated with that port number from Table 2
      iv. If the port number is not included in Table 2, return the string 'Protocol: Unknown - Unknown'

   g. Write a function named secure that:
      i. Takes one argument: a valid port number (integer)

ii. Returns the string “Secure: [True/False]” where True/False is the secure status associated with that port number from Table 2
iii. If the port number is not included in Table 2, return the string ‘Secure: Unknown’

h. Write a function named main that:
i. Takes no arguments
ii. Prompts the user for a port number with the string ‘Enter port number: ’; assume the user will enter an integer
iii. If the user enters an invalid port number (Table 1), prints “Invalid port number”
iv. If the user enters a valid port number (Table 1), prints the following information:
- Port: [PORT_NUMBER]
- Range: [RANGE_NAME]
- Category: [CATEGORY]
- Protocol: [ACRONYM/SHORT_NAME] - [LONG/FULL_NAME]
- Secure: [TRUE/FALSE]

4. Block Diagram

Draw a block diagram in the below box that describes the control flow of the function main that incorporates the other functions that you wrote.

5. Grading policy

1. Correctness: Does the program run correctly and follow the requirements? (50 points)
2. Design: Is the design a meaningful plan to solving the problem? (20 points)
3. Style: Does the code follow the documentation and submission guidelines? (10 points)
4. Is the code well structured, readable, and indented properly? (10 points)
5. Do you use meaningful variable names? (10 points)