Program 1

Programming to C

CSCI 112, Fall 2020

Objectives

- Create arrays, use a makefile
- Create a bash script, get arguments from the command line
- Manage memory

Description/Requirements:

- A network address is made up of 4 sections – each an integer from 0-255. Each section is separated by a period (.) when shown to the user.
- Depending on the class of the address, one or more sections identify the network and one or more sections identify the host within the network. This table shows this

Class	First octet value	Subnet mask
A	0-127	8
B	128-191	16
C	192-223	24
D	224-239	-
E	240-255	-

- For the IP addresses from Class A, the first 8 bits (the first number) represent the network part, while the remaining 24 bits (last 3 numbers) represent the host part. For Class B, the first 16 bits (the first two numbers) represent the network part, while the remaining 16 bits (last 2 numbers) represent the host part. For Class C, the first 24 bits represent the network part (the first 3 numbers), while the remaining 8 bits (the last number) represent the host part.
- There are 4 input files that you will use for this program – you will run the program 4 times – once with each run. All of these files are in /public/pgm1.
- You will write a bash script to run your program 4 times. Redirect the output from the script to some output file.
- You will read in the data in the file redirected from standard input. The files required are: inp17.txt, inp2000.txt, inp5000.txt, inp8000.txt. There are other files in the directory to help you debug such as inpa.txt which is just a few A addresses.
- You will pass to the program via the command line the number of addresses in each file. Example: ./pgm1 17 < inp17.txt
- NOTE: We are using traditional network addresses (not classless addresses (CIDR)).

- You must have at least 3 functions besides main. Each function must be in a separate file. You will use a makefile to compile each source file separately and link the resulting objects files together to create one executable.
- **What the program will do:**
- Convert the number of addresses from the command line to an int. Print an error and end the program if you there is not one argument plus the executable name on the command line.
- You will read each address into a 2 dimensional array of unsigned chars (that is to minimize the amount of memory you use since all numbers are between 0 and 255). The size of the array is known by the value you passed in through the command line.
- Sort the array.
- Count the number of networks in each class for classes A, B, and C. Don't count any addresses that are class D or class E.
- Determine if there is more than 1 host in each network.
- Print the number of networks in each class.
- Print the network address that has the most hosts and the number of hosts in that network.
- DO NOT USE GLOBALS.
- MUST COMPILE WITH -Wall
- You must submit:
- 1) Screen shot showing your successful makefile run and the successful script run.
- 2) source code
- 3) the output file (the one you redirected output into for the 4 runs)
- 4) makefile
- Hint: to read in an unsigned char you use %hhu

My Output

```
[k57h721@csci112 pgm1]$ ./runs.sh > out.txt
[k57h721@csci112 pgm1]$ cat out.txt
running pgm1 with 17 addresses
Class A has 5 networks
Largest A network is 106 with 3 hosts
Class B has 5 networks
Largest B network is 137.249 with 2 hosts
Class C has 2 networks
Largest C network is 215.116.26 with 2 hosts

running pgm1 with 2000 addresses
Class A has 128 networks
Largest A network is 38 with 16 hosts
Class B has 484 networks
Largest B network is 129.74 with 2 hosts
Class C has 256 networks
All C networks have only 1 host

running pgm1 with 5000 addresses
Class A has 128 networks
Largest A network is 95 with 35 hosts
Class B has 1234 networks
Largest B network is 144.40 with 3 hosts
Class C has 640 networks
All C networks have only 1 host

running pgm1 with 8000 addresses
Class A has 128 networks
Largest A network is 106 with 50 hosts
Class B has 1961 networks
Largest B network is 141.175 with 4 hosts
Class C has 987 networks
Largest C network is 216.24.49 with 2 hosts
```

Submission

- Due Date: Sunday, 10/4 at 11pm

  Each student will complete and submit this assignment individually. I will check for plagerism. Labs submitted after the due date/time will not be accepted.

Grading

Points (100 pts) – IF YOU DO IT ALL CORRRECTLY BUT ONLY USE ONE SOURCE AND NO MAKEFILE AND NO BASH SCRIPT YOU WILL GET AN 85

- 5 points – comments explaining what your program does
- 15 points – submitted screenshot, output file and source code as required above
- 10 points – meets requirements above that I didn't specify below – uses a command line parameter, has error msg if no command line parameter, 2 dim array is of size: command line number x 4.
- 20 points – compiles successfully with -Wall – no warnings
- 5 points – ran the 4 runs with one bash script
- 5 points – used a makefile
- 5 points – had each function in a separate file
- 5 points – redirect the input into the program
- 10 points – printed the information specified in the requirements
- 10 points – read all the data into a 2 dimensional array
- 10 points – sorted the addresses
- EXTRA CREDIT: 5 points if you use a sort other than selection sort (which is what I provided in class)