Hacettepe University
Computer Science
and Engineering Department

Name and Surname	**:** ATAKAN AYYILDIZ
Identity Number	**:** 21526681
Course	**:** BBM203 PROGRAMMING LAB
Subject	**:** Stack, Queue, and Dynamic Memory Allocation
Experiment	:Gain knowledge on C
Data Due	**: 25.11.2018 (23:59:59)**
Main Program	**:HUBBMNET**

PROBLEMS:

1. Using struct data structures passing datas to the stacks and queue build a C Project.
2. Most important problem is creating dynamic array structs and free them properly.
3. While datas pass to the structures stacks and queues size should increase one by one.
4. Same operations for the pop functions.
5. Reading dynamic files passing them to the structure arrays or 2D array.
6. Reading commands line by line operate them.
7. After free the pointers array structures and close the files.
8. Write a recursive function to send message.

SOLUTİONS:

1. Stack(Frame) is creating to store datas with int top and char double pointer array variables.Push pop and peek function add-delete and show the frame datas.
2. There are two queues(inqueue-outqueue) for each client and they are stores frames.And there is a enqueue function to pass frames to the queues.
3. Struct PC structures stores clients, infos, queues and hops.
4. Files are reading…
5. Clients info passing from file to the struct PC *clientsArray[i]
6. 2D Intended array stores Intended Destination
7. 2D nextarray stores Neighbor to which the packet should be forwarded

8. Command.dat processing…
9. While iterate until reach the commandNumber
10. If the first command Message firstly use strtok to parse string from (#).
11. Calcute frameNumber and parsing message and passing to the 2D parted message array
12. Find the correct client and pass the datas to the frames one by one with push function.
13. Print the datas to the screen with pop and peek functions and push them again.
14. If the command SHOW_FRAME_INFO first control if the wanted frame exist otherwise print No such frame.
15. Else find the intended queue of the client and print datas to the screen.
16. If the command SHOW_Q_INFO find the client's queue and print the frame numbers.
17. If the command is SEND go to Send recursive function with written arguments.
18. Firstly find where the tempSender for example first temp. Sender C then B it goes to D >> E.
19. After finding temp sender then we search what is the final intended direction then we control there is no '-' then we search the B as second sender then we serch 3.rd sender D and control it does not "-" if it "-" we print an error message.
20. Also second sender will be first receiver and we realloc first receiver's inqueue then passing frames to first receiver's inqueue then frames directly go to the same client's outqueue.
21. This will continue in the recursion function until frames reach the final direction.

22. If the command PRINT_LOG using time.h library and find the current time and search if the frames are in queues of clients by calculating frames top variable print to the screen.
23. If the command is not appropriate with theese commands print Invalid command to the screen.
24. Finally free all the dynamic arrays and structures.
25. Close the files.

MAIN DATA STRUCTURES:
Stacks and queues and clients infos are build with structs.

Stacks have int top variables it stores how many datas are in frames. 2D Data array stores infos of layers.

Queues are the same have rear variables same as top. And front variable where is the first frame.Struct Stack frame holds frames.

After all memory checked with valgrind.And fix some problems of reallocation and invalid read-write.