Whatsapp Mimic in C Program.  
Program:

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#define MAX\_SIZE 100

int top = -1;

int top2 = -1;

int stack[MAX\_SIZE];

int readMsg1 = 0;

int readMsg2 = 0;

char \*\*arrayMsgs1;

char \*\*arrayMsgs2;

void push1(char \*message)

{

arrayMsgs2[++top2] = message;

return;

}

void push2(char \*message);

void unreadMessages2();

char \*allocateMemoryForMessage(char \* msg)

{

return (char \* ) malloc(strlen(msg) \* sizeof(char) + 1);

}

void sendMessageTo2()

{

char messageToBeSent[100];

sendAnotherMessage:

printf("\nEnter your message: ");

scanf(" %[^\n]%\*c", messageToBeSent);

char \* msgBuffer = allocateMemoryForMessage(messageToBeSent);

strcpy(msgBuffer, messageToBeSent);

push1(msgBuffer);

printf("\nWant to send another message?(Y/N)");

if (toupper(getch()) == 'Y')

{

goto sendAnotherMessage;

}

return;

}

void unsendTheLastMessage1()

{

if (top2 == -1)

{

printf("\nNo messages sent");

}

else

{

printf("\nDeleting the message: %s", arrayMsgs2[top2]);

arrayMsgs2[top2] = NULL;

free(arrayMsgs2[top2--]);

}

return;

}

void peek1()

{

if (top == -1)

{

printf("\nNothing to peek upon.");

}

else

{

printf("The element at the top is %d", stack[top]);

}

return;

}

void isempty1()

{

if (top == -1)

printf("\nThe stack is empty.");

else

printf("\nThe stack is not empty.");

return;

}

void displayAllMessages1()

{

if (top == -1)

{

printf("\nNo messages to display");

return;

}

int i;

for (i = 0; i <= top; i++)

{

printf("\n%s", arrayMsgs1[i]);

}

readMsg1 = top + 1;

return;

}

void displayAllMessagesAsAChat1()

{

int i;

for(i=0;i<=top)

}

void unreadMessages1()

{

if (readMsg1 > top)

{

printf("\nNo messages unread");

return;

}

printf("\n%d Unread Messages", top - readMsg1 + 1);

while (readMsg1 <= top)

{

printf("\n%s", arrayMsgs1[readMsg1]);

readMsg1++;

}

}

void sendMessageTo1()

{

char messageToBeSent[100];

sendAnotherMessage:

printf("\nEnter your message: ");

scanf(" %[^\n]%\*c", messageToBeSent);

char \* messageBuf = allocateMemoryForMessage(messageToBeSent);

strcpy(messageBuf, messageToBeSent);

push2(messageBuf);

printf("\nSend another message?(Y/N)");

if (toupper(getch()) == 'Y')

{

goto sendAnotherMessage;

}

}

void push2(char \* message)

{

arrayMsgs1[++top] = message;

return;

}

void unsendTheLastMessage2()

{

if (top == -1)

{

printf("\nNo messages sent");

}

else

{

printf("\nDeleting the message: %s", arrayMsgs1[top]);

arrayMsgs1[top] = NULL;

free(arrayMsgs1[top--]);

}

return;

}

void peek2()

{

if (top2 == -1)

{

printf("\nNothing to peek upon.");

}

else

{

printf("The element at the top is %d", stack[top2]);

}

return;

}

void isempty2()

{

if (top2 == -1)

printf("\nThe stack is empty.");

else

printf("\nThe stack is not empty.");

}

void displayAllMessages2()

{

if (top2 == -1)

{

printf("\nNo messages to display");

return;

}

int i;

for (i = 0; i <= top2; i++) {

{

printf("\n%s", arrayMsgs2[i]);

}

readMsg2 = top + 1;

return;

}

void unreadMessages2()

{

if (readMsg2 > top2)

{

printf("\nNo messages unread");

return;

}

printf("\n%d Unread Message(s)", top2 - readMsg2 + 1);

while (readMsg2 <= top2)

{

printf("\n%s", arrayMsgs2[readMsg2]);

readMsg2++;

}

}

int main()

{

arrayMsgs1 = (char \*\* ) malloc(100 \* sizeof(char \* ));

arrayMsgs2 = (char \*\* ) malloc(100 \* sizeof(char \* ));

int person;

personSelection:

printf("\nWhich person are you?(1 or 2) ");

scanf("%d", & person);

if (person != 1 && person != 2)

{

printf("\nSelect a valid person");

goto personSelection;

}

int choice;

do

{

printf("\n\n-----Menu-----");

printf("\n1:Send a message");

printf("\n2:Read unread messages (%d)", person == 1 ? top - readMsg1 + 1 : top2 - readMsg2 + 1);

printf("\n3:Display all the messages received");

printf("\n4:Unsend the last message");

printf("\n5:Switch Person");

printf("\n6:Exit");

printf("\n\nEnter your choice\n");

scanf("\n%d", & choice);

switch (choice)

{

case 1:

printf("\nEnter the message to send: \n");

if (person == 1)

sendMessageTo2();

else

sendMessageTo1();

break;

case 2:

if (person == 1)

unreadMessages1();

else

unreadMessages2();

break;

case 3:

if (person == 1)

unsendTheLastMessage1();

else

unsendTheLastMessage2();

break;

case 4:

person == 1 ? unsendTheLastMessage1() : unsendTheLastMessage2();

break;

case 5:

if (person == 1)

{

printf("\nSwitching to the second person.");

person = 2;

}

else

{

printf("\nSwitching to the first person.");

person = 1;

}

break;

case 6:

printf("\nExited ");

break;

default:

printf("\nPlease enter a value between 1 and 4");

break;

}

}while (choice != 6);

int i, j;

for (i = 0, j = 0; i <= top, j <= top2; i++, j++)

{

free(arrayMsgs1[i]);

free(arrayMsgs2[j]);

}

return 0;

}