historySearch.c

```
#include "defs.h"
#include
#include
#include

//GROUP A
//Name: Okonkwo, Thomas Chukwuma
//CWID: A20308887
//Email: thomas.okonkwo@okstate.edu

bool equalsIgnoreCase(char* str1, char* str2){

//compares the length of two strings passed
if (strlen(str1) != strlen(str2)){
return FALSE;
}else{
//loops through the string and convert to lower case
//so that we can compare each letter of the strings
for (int i = 0; i < strlen(str1); i++){
if (tolower(str1[i]) != tolower(str2[i])){
return FALSE;
}
}
}
return TRUE;
}

//this function searches the text file to see if a string is present in it
bool historySearch(char *query){
//we read from the history file
FILE* infile = fopen("History.txt","r");

//created string variables/placeholders for the split of the query string
//because we will have to search with the query string to see if the content
//of the query is present in the history file. SO it's best practice to split
//the query string into different string variables so we can search the history
//file easily with these new variables.
char nameQuery[256] = {0}, jobQuery[256] = {0}, statusQuery[256] = {0};
char job[256] = {0}, name[256] = {0}, status[256] = {0};
char buffer[1024] = {0};
bool foundMatch = FALSE;
float id = 0;
int count = 0;

//since in the original file, it is comma seperated, I decided to format
//the string query into 3 different string variable names. This is so we
//can search the history file with each of these string variables instead
//of using the query string.
```

```
sscanf(query, "%[^,],%[^,],%[^,]", nameQuery, jobQuery, statusQuery);

//created a temp buffer that copies the content of the history file read
while (fgets(buffer, 1024, infile)){
//so we split the buffer content as well in to name, job and status so we can use
//them to compare with the query string variable(split in 3 as shown above).
sscanf(buffer, "%*d,%[^,],\"%[^\"]\",%*f,%*f,%*f,%[^,],%*
[^\n\r]",name,job,status);

//compares the job string to see if it's empty and format the main string into
//name, job and status.
if (strcmp(job, "") == 0){
sscanf(buffer, "%*d,%[^,],%[^,],%*f,%*f,%*f,%[^,],%*[^\n\r]",name,job,status);

}
//compares and checks for the case sensitivity and ignores case sensitivity
if (equalsIgnoreCase(name, nameQuery) && equalsIgnoreCase(job, jobQuery) &&
equalsIgnoreCase(status, statusQuery)){
termPrinter("Record found in File");
termPrinter(buffer); //passed to the termPrinter function
return TRUE;
}
strcpy(job, "");
}

fclose(infile); //closed the file
return FALSE;
}
```