```c
// Author: John Miller
// Email: john.t.miller@okstate.edu

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <stdbool.h>
#include "Assistant.h"
#include "History.h"
#include "Query.h"
#include "Record.h"

#define MAX_HISTORY_CHARS 500
#define MAX_HISTORY_LINES 10

void getFromHistory(Query search, Record *result){
  Record found;

  FILE *historyFile;
  historyFile = fopen("history.txt", "r");
  char line[MAX_HISTORY_CHARS];

  if(historyFile){
    while(fgets(line, sizeof(line), historyFile)){

        //split line into array of strings on commas
        int s=0;
        char *nextTokenPnt = strtok(line, "\t");
        char *substrings[14];
        while(nextTokenPnt != NULL){
            substrings[s] = nextTokenPnt;
            s++;
            nextTokenPnt = strtok(NULL, "\t");
        }

        //printf("%s, %s, %s, %s\n",substrings[0], substrings[1], substrings[2], substrings[6]);
        if(strcmp(substrings[1], search.employee_name)==0 && strcmp(substrings[2], search.job_title)==0 && strcmp(substrings[6], search.status)==0){ //employee is in history
            found.id = atoi(substrings[0]);
            strcpy(found.employee_name,substrings[1]);
            strcpy(found.job_title,substrings[2]);
            found.base_pay = strtof(substrings[3],NULL);
            found.overtime_pay = strtof(substrings[4],NULL);
            found.benefit = strtof(substrings[5],NULL);
            strcpy(found.status, substrings[6]);
            found.satisfaction_level = strtof(substrings[7],NULL);
            found.number_project = atoi(substrings[8]);
            found.average_monthly_hours = atoi(substrings[9]);
            found.time_spend_company_in_yrs = atoi(substrings[10]);
            found.work_accident = atoi(substrings[11]);
            found.promotion_last_5years = atoi(substrings[12]);

            break;
        }
    }
  }
  else
  {
    printf("Error Reading File: history.txt\n");
  }

  fclose(historyFile);
  *result = found;
}
```

```c
bool isInHistory(Query search){
  FILE *historyFile;
  historyFile = fopen("history.txt", "r");
  char line[MAX_HISTORY_CHARS];
  bool found = false;
  char *rest = line;

  if(historyFile){
    while(fgets(line, sizeof(line), historyFile)){


        //split line into array of strings on commas
        int s=0;
        char *nextTokenPnt = strtok_r(line, "\t", &rest);
        char *substrings[13];

        while(nextTokenPnt != NULL){
            substrings[s] = nextTokenPnt;
            s++;
            nextTokenPnt = strtok_r(NULL, "\t", &rest);
        }

        // printf("%s,%s,%s,%s\n",substrings[0], substrings[1], substrings[2], substrings[6]);
        // printf("search: employeename: %s jobtitle %s status %s", search.employee_name, search.job_title, search.status);
        if(strcmp(substrings[1], search.employee_name)==0 && strcmp(substrings[2], search.job_title)==0 && strcmp(substrings[6], search.status)==0){ //employee is in history
            found = true;
            break;
        }

    }
  }
  else{
    printf("Error Reading File: history.txt\n");
  }

  fclose(historyFile);

  return found;
}

void writeToHistory(Record *record){
    char history[MAX_HISTORY_LINES+1][MAX_HISTORY_CHARS];

    char writeLine[MAX_HISTORY_CHARS];

    sprintf(writeLine,"%d\t%s\t%s\t%f\t%f\t%f\t%s\t%f\t%d\t%d\t%d\t%d\t%d\n", record->id,record->employee_name,record->job_title,record->base_pay,record->overtime_pay,record->benefit,record->status,record->satisfaction_level,record->number_project,record->average_monthly_hours,record->time_spend_company_in_yrs,record->work_accident,record->promotion_last_5years);
    //strcpy(writeLine, "TESTING\n");
    // printf("Writing to history:\n%s\n", writeLine);

    FILE *historyFile;
    historyFile = fopen("history.txt", "r");
    char line[MAX_HISTORY_CHARS];

    int lineNumber = 0;

    if(historyFile){
        while(fgets(line,sizeof(line), historyFile)){
        strcpy(history[lineNumber],line);
        lineNumber++;
        }

        fclose(historyFile);

        if(lineNumber<MAX_HISTORY_LINES){//just append
            strcpy(history[lineNumber],writeLine);
            lineNumber++;
            //write to file 0-9
            FILE *writeHistoryFile;
            writeHistoryFile = fopen("history.txt", "w+");

            if(writeHistoryFile){
                for(int i=0; i<lineNumber; i++){
                    fprintf(writeHistoryFile,"%s",history[i]);
                }
            }
            else{
                printf("Error Writing File: history.txt\n");
            }
            fclose(writeHistoryFile);
        }
        else{
            strcpy(history[MAX_HISTORY_LINES],writeLine);
            //write to file 1-10
            FILE *writeHistoryFile;
            writeHistoryFile = fopen("history.txt", "w+");

            if(writeHistoryFile){
                for(int i=1; i<=MAX_HISTORY_LINES; i++){
                    fprintf(writeHistoryFile,"%s",history[i]);
                }
            }
            else{
                printf("Error Writing File: history.txt\n");
            }
            fclose(writeHistoryFile);
        }
    }
    else
    {
       printf("Error Reading File: history.txt\n");
       fclose(historyFile);
    }

```