**Advanced Procedural Programming Assignment**

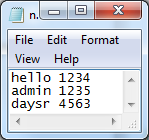
By Andrei Petruk

G00328108

I implement the survey database as a linked list.

Main function includes code for username and password recognition which stored in name\_pass.txt.

And menu to manipulate the linked list:

1. Add survey

Function to add new element, this function also reused by edit by pps function.

I use " %[^\n]s" to enter address with spaces.

Strstr to check if e-mail contains @ and .com.

1. Display all surveys to screen

Function to display the contents of the list to screen also reused in searching function.

1. Search by Second name

Takes Second name and search for it by comparing stored names in list

Then reuse function “Display all surveys to screen” to display information about particular person.

1. Delete a survey from the list by PPS number.

Searching for PPS and removes node.

This function has 3 functions inside: one to delete top node, other to delete last node and counting function.

1. Update a survey in list by PPS number.

Also Searching for PPS and updates survey using questions from add survey function.

1. Generate statistic and print to file.

This function generates statistics and print to statistic file Statistics.text

1. Output all of the survey in the list to file.

This function write the contents of the list to file Survey.text

1. Save all of the survey in the list to file
2. Load the surveys from the file

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include <string.h>

typedef struct //Define structure for data

{ //Variables Details of person

int ppsNum;

char firstName[30];

char secondName[30];

char address[50];

char email[30];

//Groops

int gender;

int age;

int income;

//Survey questions

int exersise;

int alcohol;

int cigarettes;

}Survey\_t;

// Definition of the node structure for the linked list

struct listelement //define linked list structure

{

Survey\_t data; //structured data

struct listelement \*next; //poiter to the next node

};

// Definition of the various functions used to manipulate the linked list

void addSurvey(struct listelement\*\* head\_ptr, int pPnum, struct listelement \*temp);

void printSurveys(struct listelement\*\* head\_ptr, char surename[20]);

void search(struct listelement \*head\_ptr, char surename[20]);

void deleteAtPos(struct listelement\*\* head\_ptr, int pPnum);

void deleteTopNode(struct listelement\*\* head\_ptr);

int deleteNode(struct listelement\*\* head\_ptr);

int countSurv(struct listelement\* head\_ptr);

void editAtPos(struct listelement \*head\_ptr, int pPnum);

void fileOutput(struct listelement\*\* head\_ptr);

void statistics(struct listelement\* head\_ptr);

void fileOutputSave(struct listelement\* head\_ptr);

void fileLoadSave(struct listelement\*\* head\_ptr);

int main() {

//Declare all variables --- pointer to top of the list,

//choice of function by the user, string being

// Pointer to file

FILE\* filep;

struct listelement\* top;

int num;

int pPnum = 999999;

int choice;

char surename[20], username[15], password[15], ch = 0, usr[15], pass[15];

int i = 0, found = 0;

// List is empty at the start

top = NULL;

//open file

filep = fopen("name\_pass.txt", "r");

if (filep == NULL)

{

printf("File Cannot be opened \n");

}

printf("Username: "); //user input username and password//

scanf("%s", usr);

printf("Password: ");

while (ch != 13)

{

ch = getch();

pass[i] = ch;

i++;

printf("\*");

}

pass[i] = '\0';

while (!feof(filep) && found == 0)

{

fscanf(filep, "%s %s ", username, password); //scan from file

if (strcmp(usr, username) == 1 && strcmp(pass, password) == 1)

found = 1;

if (found == 1) {

printf("\nInvalid username or password!");

printf("\nPlease enter again!");

printf("\n");

main();

}

}

printf("\n Welcome! ");

fclose(filep);

// Display user menu

do {

printf("\n ");

printf("\n 1. Add an survey: \n");

printf(" 2. Print all surveys: \n");

printf(" 3. Search by Second name: \n");

printf(" 4. Delete a survey from the list by PPS number: \n");

printf(" 5. Update a survey in list by PPS number: \n");

printf(" 6. Generate statistic and print to file: \n");

printf(" 7. Print all of the survey in the list to file. \n");

printf(" 8. Save all of the survey in the list to file. \n");

printf(" 9. Load the surveys from the file. \n");

printf("-1. Exit\n\n");

printf("Please enter your choice: ");

scanf("%d", &choice);

if (choice == 1) {

addSurvey(&top, pPnum, top);

}

else if (choice == 2) {

printSurveys(top, surename);

}

else if (choice == 3) {

printf("\n Enter Person Second Name you sershing for: ");

scanf("%s", surename);

search(top, surename);

}

else if (choice == 4) {

printf("Please enter PPS to delete survey: ");

scanf("%d", &pPnum);

deleteAtPos(&top, pPnum);

pPnum = 999999;

}

else if (choice == 5) {

printf("Please enter PPS to update survey: ");

scanf("%d", &pPnum);

editAtPos(&top, pPnum);

pPnum = 999999;

}

else if (choice == 6) {

statistics(top);

}

else if (choice == 7) {

fileOutput(top);

}

else if (choice == 8) {

fileOutputSave(top);

}

else if (choice == 9) {

fileLoadSave(&top);

}

} while (choice != -1);// Continue to allow user input until -1 is entered

}

// Function to add an element

void addSurvey(struct listelement\*\* head\_ptr, int pPnum, struct listelement \*temp) {

struct listelement \*newNode;

if (pPnum == 999999)

{ //Create the new node

newNode = (struct listelement\*)malloc(sizeof(struct listelement));

}

else

{

newNode = temp;

}

if (pPnum == 999999)

{

printf("\nPlease enter pps number: ");

scanf("%d", &newNode->data.ppsNum);

}

printf("\nPlease enter first name: ");

scanf("%s", newNode->data.firstName);

printf("\nPlease enter second name: ");

scanf("%s", newNode->data.secondName);

printf("\nPlease enter e-mail: ");

scanf("%s", newNode->data.email);

//check if Email contains "@" and .com

while (strstr(newNode->data.email, "@") == 0 || strstr(newNode->data.email, ".com") == 0)

{

printf("\Email Address must contain an @, .com : \n");

printf("\nPlease re enter e-mail: ");

scanf("%s", newNode->data.email);

}

printf("\nPlease enter Gender: (Press: 1 for male) (Press: 2 for female) : ");

scanf("%d", &newNode->data.gender);

printf("\nPlease enter Age Bracket: ");

printf("\n\tPress 1 if your age: 18-20 yrs");

printf("\n\tPress 2 if your age: 20-30 yrs");

printf("\n\tPress 3 if your age: 30-50 yrs");

printf("\n\tPress 4 if your age: 50-65 yrs");

printf("\n\tPress 5 if your age: 65+ yrs\n");

scanf("%d", &newNode->data.age);

printf("\nPlease enter Income Bracket: ");

printf("\n\tPress 1 if No Income ");

printf("\n\tPress 2 if Less than 20,000");

printf("\n\tPress 3 if Less than 40,000");

printf("\n\tPress 4 if Less than 60,000");

printf("\n\tPress 5 if Less than 80,000");

printf("\n\tPress 6 if Less than 100,000");

printf("\n\tPress 7 if Greater than 100,000\n");

scanf("%d", &newNode->data.income);

printf("\nHow often do you exercise?: ");

printf("\n\tPress 1 if Never");

printf("\n\tPress 2 if Less than three times per week");

printf("\n\tPress 3 if Less than five times per week");

printf("\n\tPress 4 if More than five times per week\n");

scanf("%d", &newNode->data.exersise);

printf("\nHow much alcohol do you consume per week?: ");

printf("\n\tPress 1 if None");

printf("\n\tPress 2 if Less than 2 units");

printf("\n\tPress 3 if Less than 4 units");

printf("\n\tPress 4 if More than 4 units\n");

scanf("%d", &newNode->data.alcohol);

printf("\nHow many cigarettes do you smoke per week?: ");

printf("\n\tPress 1 if None");

printf("\n\tPress 2 if Less than 20 cigarettes");

printf("\n\tPress 3 if Less than 40 cigarettes");

printf("\n\tPress 4 if More than 40 cigarettes\n");

scanf("%d", &newNode->data.cigarettes);

printf("\nPlease enter address: ");

scanf(" %[^\n]s", newNode->data.address);

if (pPnum != 999999)

{

return;

}

else

{ //Connect up the node so that the new node is the headpointer

newNode->next = \*head\_ptr;

\*head\_ptr = newNode; // Transfer the address of newNode to head

}

}

// Function to display the contents of the list

void printSurveys(struct listelement\* head\_ptr, char surename[20]) {

int data;

char surenameDef[20];

struct listelement \*temp;

temp = head\_ptr;

printf("Survey details: \n");

while (temp != NULL) { // Go to the last node

///Display the contents

printf("\n PPSno: %d ", temp->data.ppsNum); // show the data

printf("\n First Name: %s ", temp->data.firstName);

printf("\n Second Name: %s ", temp->data.secondName);

printf("\n E-mail: %s ", temp->data.email);

if (temp->data.gender == 1)

printf("\n Gender: male");

else

printf("\n Gender: female");

if (temp->data.age == 1)

printf("\n Age Bracket: 18-20 yrs");

else if (temp->data.age == 2)

printf("\n Age Bracket: 20-30 yrs");

else if (temp->data.age == 3)

printf("\n Age Bracket: 30-50 yrs");

else if (temp->data.age == 4)

printf("\n Age Bracket: 50-65 yrs");

else if (temp->data.age == 5)

printf("\n Age Bracket: 65+ yrs");

if (temp->data.income == 1)

printf("\n No Income");

else if (temp->data.income == 2)

printf("\n Income Less than 20,000");

else if (temp->data.income == 3)

printf("\n Income Less than 40,000");

else if (temp->data.income == 4)

printf("\n Income Less than 60,000");

else if (temp->data.income == 5)

printf("\n Income Less than 80,000");

else if (temp->data.income == 6)

printf("\n Income Less than 100,000");

else if (temp->data.income == 7)

printf("\n Income Greater than 100,000");

if (temp->data.exersise == 1)

printf("\n Never exercise");

else if (temp->data.exersise == 2)

printf("\n Exercise Less than three times per week");

else if (temp->data.exersise == 3)

printf("\n Exercise Less than five times per week");

else if (temp->data.exersise == 4)

printf("\n Exercise More than five times per week");

if (temp->data.alcohol == 1)

printf("\n No alcohol ");

else if (temp->data.alcohol == 2)

printf("\n Less than 2 units of alcohol per week");

else if (temp->data.alcohol == 3)

printf("\n Less than 4 units of alcohol per week");

else if (temp->data.alcohol == 4)

printf("\n More than 4 units of alcohol per week");

if (temp->data.cigarettes == 1)

printf("\n No cigarettes ");

else if (temp->data.cigarettes == 2)

printf("\n Less than 20 cigarettes per week");

else if (temp->data.cigarettes == 3)

printf("\n Less than 40 cigarettes per week");

else if (temp->data.cigarettes == 4)

printf("\n More than 40 cigarettes per week");

printf("\n Address: %s ", temp->data.address);

printf("\n");

if (strcmp(surename, temp->data.secondName) == 0)

{

strcpy(surename, surenameDef);

return;

}

else

{

temp = temp->next; // move to next node

}

}

}

// Function to search by second name

void search(struct listelement \*head\_ptr, char surename[20])

{

struct listelement \*temp;

int count = 0;

temp = head\_ptr;

while (temp != NULL)

{

if (strcmp(temp->data.secondName, surename) == 0)//comparing strings

{

printf("\nStudent ID %s found at position %d\n", surename, (count + 1));

//reuse of printSurveys function but only for specific Second name

printSurveys(temp, surename);

return;

}

temp = temp->next;

count++;

}

printf("\nData not found");

}

//This function deletes a node by pps

void deleteAtPos(struct listelement\*\* head\_ptr, int pPnum) {

struct listelement \*prev\_ptr, \*cur\_ptr, \*temp;

int i;

int count = 0;

cur\_ptr = \*head\_ptr;

temp = \*head\_ptr;

while (temp != NULL)

{

if (temp->data.ppsNum == pPnum)

{

count++;

printf("\nPPS %d found at position %d\n", pPnum, count);

if (count == 1) {

//call function to dalete top node

deleteTopNode(head\_ptr);

return;

}

else if (count == countSurv(\*head\_ptr)) {

//call function to dalete last node

deleteNode(head\_ptr);

return;

}

else {

for (i = 1; i<count; i++)

{

prev\_ptr = cur\_ptr;

cur\_ptr = cur\_ptr->next;

}

prev\_ptr->next = cur\_ptr->next;

printf("\n Survey deleted.");

free(cur\_ptr);

return;

}

return;

}

temp = temp->next;

count++;

}// end of while

printf("\nData not found");

}

//function to dalete top node used by deleteAtPos

void deleteTopNode(struct listelement\*\* head\_ptr) {

struct listelement \*temp;

if (\*head\_ptr == NULL) {

printf("Nothing to delete\n");

}

else {

temp = (struct listelement\*)malloc(sizeof(struct listelement));

temp = \*head\_ptr;

\*head\_ptr = temp->next;

printf("\n Survey deleted.");

free(temp);

}

}

//function to dalete last node used by deleteAtPos

int deleteNode(struct listelement\*\* head\_ptr)

{

int data;

struct listelement \*temp;

struct listelement \*newNode;

struct listelement \*old\_temp;

if (\*head\_ptr == NULL) {

printf("Nothing to delete\n");

}

else {

temp = (struct listelement\*)malloc(sizeof(struct listelement));

temp = \*head\_ptr;

old\_temp = (struct listelement\*)malloc(sizeof(struct listelement));

while (temp->next != NULL) {

old\_temp = temp;

temp = temp->next;

}

old\_temp->next = NULL;

printf("\n Survey with PPS %d deleted.", &temp->data.ppsNum);

free(temp); // free the old value

}

}

//function to count nodes used by deleteAtPos

int countSurv(struct listelement\* head\_ptr)

{

struct listelement \*n;

int c = 0;

n = head\_ptr;

while (n != NULL)

{

n = n->next; // move to next node

c++; // each time there is a non-null element, count goes up

}

//printf("Number of surveys in the list is: %d\n", c);

return c; // number of elements in the list

}

//This edit a node by pps

void editAtPos(struct listelement\*\* head\_ptr, int pPnum)

{

struct listelement \*temp;

int count = 0;

temp = \*head\_ptr;

while (temp != NULL)

{

if (temp->data.ppsNum == pPnum)

{

printf("\nStudent ID %d found at position %d\n", pPnum, (count + 1));

//reuse of addSurvey function to edit details

addSurvey(head\_ptr, pPnum, temp);

return;

}

temp = temp->next;

count++;

}

printf("\nData not found");

}

// This function write the contents of the list to file

void fileOutput(struct listelement\* head\_ptr)

{

int data;

FILE\* theFile;

errno\_t err;

struct listelement \*temp;

temp = head\_ptr;

//Open the file to write...

if ((err = fopen\_s(&theFile, "Survey.text", "w")) != 0)

printf("Error opening file Survey.text for writing!");

else

{

fprintf\_s(theFile, "Survey detailss: ");

while (temp != NULL)// Go to the last node

{

//output the contents to file if it was opened

fprintf(theFile, "\n PPSno: %d ", temp->data.ppsNum); // show the data

fprintf(theFile, "\n First Name: %s ", temp->data.firstName);

fprintf(theFile, "\n Second Name: %s ", temp->data.secondName);

fprintf(theFile, "\n E-mail: %s ", temp->data.email);

if (temp->data.gender == 1)

fprintf(theFile, "\n Gender: male");

else

fprintf(theFile, "\n Gender: female");

if (temp->data.age == 1)

fprintf(theFile, "\n Age Bracket: 18-20 yrs");

else if (temp->data.age == 2)

fprintf(theFile, "\n Age Bracket: 20-30 yrs");

else if (temp->data.age == 3)

fprintf(theFile, "\n Age Bracket: 30-50 yrs");

else if (temp->data.age == 4)

fprintf(theFile, "\n Age Bracket: 50-65 yrs");

else if (temp->data.age == 5)

fprintf(theFile, "\n Age Bracket: 65+ yrs");

if (temp->data.income == 1)

fprintf(theFile, "\n No Income");

else if (temp->data.income == 2)

fprintf(theFile, "\n Income Less than 20,000");

else if (temp->data.income == 3)

fprintf(theFile, "\n Income Less than 40,000");

else if (temp->data.income == 4)

fprintf(theFile, "\n Income Less than 60,000");

else if (temp->data.income == 5)

fprintf(theFile, "\n Income Less than 80,000");

else if (temp->data.income == 6)

fprintf(theFile, "\n Income Less than 100,000");

else if (temp->data.income == 7)

fprintf(theFile, "\n Income Greater than 100,000");

if (temp->data.exersise == 1)

fprintf(theFile, "\n Never exercise");

else if (temp->data.exersise == 2)

fprintf(theFile, "\n Exercise Less than three times per week");

else if (temp->data.exersise == 3)

fprintf(theFile, "\n Exercise Less than five times per week");

else if (temp->data.exersise == 4)

fprintf(theFile, "\n Exercise More than five times per week");

if (temp->data.alcohol == 1)

fprintf(theFile, "\n No alcohol ");

else if (temp->data.alcohol == 2)

fprintf(theFile, "\n Less than 2 units of alcohol per week");

else if (temp->data.alcohol == 3)

fprintf(theFile, "\n Less than 4 units of alcohol per week");

else if (temp->data.alcohol == 4)

fprintf(theFile, "\n More than 4 units of alcohol per week");

if (temp->data.alcohol == 1)

fprintf(theFile, "\n No alcohol ");

else if (temp->data.alcohol == 2)

fprintf(theFile, "\n Less than 2 units of alcohol per week");

else if (temp->data.alcohol == 3)

fprintf(theFile, "\n Less than 4 units of alcohol per week");

else if (temp->data.alcohol == 4)

fprintf(theFile, "\n More than 4 units of alcohol per week");

if (temp->data.cigarettes == 1)

fprintf(theFile, "\n No cigarettes ");

else if (temp->data.cigarettes == 2)

fprintf(theFile, "\n Less than 20 cigarettes per week");

else if (temp->data.cigarettes == 3)

fprintf(theFile, "\n Less than 40 cigarettes per week");

else if (temp->data.cigarettes == 4)

fprintf(theFile, "\n More than 40 cigarettes per week");

fprintf(theFile, "\n Address %s ", temp->data.address);

fprintf(theFile, "\n");

temp = temp->next; // move to next node

}

printf("Data Succesfully written to Survey.text");

fclose(theFile);

}

printf("\nPress key to continue.");

\_getch();

}

//This function generates statistics and print to statistic file

void statistics(struct listelement\* head\_ptr) {

int data;

FILE\* theFile;

errno\_t err;

//counters

int cigarettesCouner[4] = { 0,0,0,0 };

int exerciseCouner[4] = { 0,0,0,0 };

int alcoholCouner[4] = { 0,0,0,0 };

int i;

struct listelement \*temp;

temp = head\_ptr;

//opens file for writing

if ((err = fopen\_s(&theFile, "Statistics.text", "w")) != 0)

printf("Error opening file Statistics.text for writing!");

else

{

printf("Statistic details: \n");

fprintf(theFile, "Statistic details: \n");

while (temp != NULL) { // Go to the last node

for (i = 0; i<4; i++)

{

if (temp->data.cigarettes == i + 1) {

cigarettesCouner[i]++;

}

if (temp->data.exersise == i + 1) {

exerciseCouner[i]++;

}

if (temp->data.alcohol == i + 1) {

alcoholCouner[i]++;

}

}

temp = temp->next; // move to next node

}

for (i = 0; i<4; i++)

{

if (cigarettesCouner[i] != 0) {

cigarettesCouner[i] = 100 / countSurv(head\_ptr)\*cigarettesCouner[i];

}

if (exerciseCouner[i] != 0) {

exerciseCouner[i] = 100 / countSurv(head\_ptr)\*exerciseCouner[i];

}

if (alcoholCouner[i] != 0) {

alcoholCouner[i] = 100 / countSurv(head\_ptr)\*alcoholCouner[i];

}

}

printf("\n %d Percent of people who do not smoke.", cigarettesCouner[0]);

printf("\n %d Percent of of people who smoke less than 20 cigarettes per week.", cigarettesCouner[1]);

printf("\n %d Percent of people who smoke less than 40 cigarettes per week.", cigarettesCouner[2]);

printf("\n %d Percent of people who smoke greater than 40 cigarettes per week.", cigarettesCouner[3]);

printf("\n %d Percent of people who never exercise.", exerciseCouner[0]);

printf("\n %d Percent of people who exercise less than three times per week.", exerciseCouner[1]);

printf("\n %d Percent of people who exercise less than five times per week.", exerciseCouner[2]);

printf("\n %d Percent of people who exercise more than five times per week.", exerciseCouner[3]);

printf("\n %d Percent of people who do not consume alcohol.", alcoholCouner[0]);

printf("\n %d Percent of people who consume less than 2 units of alcohol per week.", alcoholCouner[1]);

printf("\n %d Percent of people who consume less than 4 units of alcohol per week.", alcoholCouner[2]);

printf("\n %d Percent of people who consume more than 4 units of alcohol per week.", alcoholCouner[3]);

printf("\nData Succesfully written to Survey.text");

//print to file report

fprintf(theFile, "\n %d Percent of people who do not smoke %d", cigarettesCouner[0]);

fprintf(theFile, "\n %d Percent of of people who smoke less than 20 cigarettes per week.", cigarettesCouner[1]);

fprintf(theFile, "\n %d Percent of people who smoke less than 40 cigarettes per week.", cigarettesCouner[2]);

fprintf(theFile, "\n %d Percent of people who smoke greater than 40 cigarettes per week.", cigarettesCouner[3]);

fprintf(theFile, "\n %d Percent of people who never exercise %d", exerciseCouner[0]);

fprintf(theFile, "\n %d Percent of people who exercise less than three times per week.", exerciseCouner[1]);

fprintf(theFile, "\n %d Percent of people who exercise less than five times per week.", exerciseCouner[2]);

fprintf(theFile, "\n %d Percent of people who exercise more than five times per week.", exerciseCouner[3]);

fprintf(theFile, "\n %d Percent of people who do not consume alcohol.", alcoholCouner[0]);

fprintf(theFile, "\n %d Percent of people who consume less than 2 units of alcohol per week.", alcoholCouner[1]);

fprintf(theFile, "\n %d Percent of people who consume less than 4 units of alcohol per week.", alcoholCouner[2]);

fprintf(theFile, "\n %d Percent of people who consume more than 4 units of alcohol per week.", alcoholCouner[3]);

fclose(theFile);

}

printf("\nPress key to continue.");

\_getch();

}

void fileOutputSave(struct listelement\* head\_ptr)

{

int data;

FILE\* theFile;

errno\_t err;

struct listelement \*temp;

temp = head\_ptr;

//Open the file to write...

if ((err = fopen\_s(&theFile, "SurveySave.text", "w")) != 0)

printf("Error opening file SurveySave.text for writing!");

else

{

while (temp != NULL)// Go to the last node

{

//output the contents to file if it was opened

fprintf(theFile, "%d ", temp->data.ppsNum); // show the data

fprintf(theFile, "%s ", temp->data.firstName);

fprintf(theFile, "%s ", temp->data.secondName);

fprintf(theFile, "%s ", temp->data.email);

fprintf(theFile, "%d ", temp->data.gender);

fprintf(theFile, "%d ", temp->data.age);

fprintf(theFile, "%d ", temp->data.income);

fprintf(theFile, "%d ", temp->data.exersise);

fprintf(theFile, "%d ", temp->data.alcohol);

fprintf(theFile, "%d ", temp->data.cigarettes);

fprintf(theFile, "%s ", temp->data.address);

fprintf(theFile, "\n");

temp = temp->next; // move to next node

}

printf("Data Succesfully written to SurveySave.text");

fclose(theFile);

}

printf("\nPress key to continue.");

\_getch();

}

void fileLoadSave(struct listelement\*\* head\_ptr)

{

//int data;

FILE\* theFile;

errno\_t err;

struct listelement \*newNode, \*temp;

if ((err = fopen\_s(&theFile, "SurveySave.text", "r")) != 0)

printf("Error opening file SurveySave.text for reading!");

else

{

while (!feof(theFile))// Go to the last node

{

//output the contents to file if it was opened

newNode = (struct listelement\*)malloc(sizeof(struct listelement));

fscanf(theFile, "%d ", &newNode->data.ppsNum); // show the data

fscanf(theFile, "%s ", newNode->data.firstName);

fscanf(theFile, "%s ", newNode->data.secondName);

fscanf(theFile, "%s ", newNode->data.email);

fscanf(theFile, "%d ", &newNode->data.gender);

fscanf(theFile, "%d ", &newNode->data.age);

fscanf(theFile, "%d ", &newNode->data.income);

fscanf(theFile, "%d ", &newNode->data.exersise);

fscanf(theFile, "%d ", &newNode->data.alcohol);

fscanf(theFile, "%d ", &newNode->data.cigarettes);

fscanf(theFile, "%[^\n]s ", newNode->data.address);

if (\*head\_ptr == NULL) {

newNode->next = \*head\_ptr;

\*head\_ptr = newNode;

}

else {

temp = (struct listelement\*)malloc(sizeof(struct listelement));

temp = \*head\_ptr;

while (temp->next != NULL) {

temp = temp->next;

}

newNode->next = NULL;

if (!feof(theFile))

temp->next = newNode;

}

}//end of while

printf("\nData Succesfully loaded from SurveySave.text.");

}

fclose(theFile);

printf("\nPress key to continue.");

\_getch();

}