BLG 233E DATA STRUCTURES AND LABORATORY

EXPERIMENT 2 - FILE OPERATIONS AND POINTERS

IMPORTANT REMINDERS

1. It is not allowed to use USB sticks during the lab sessions.
2. You should unplug your ethernet cables during the lab sessions.
3. Any reference book or help material (C++) is allowed.

Write and compile the given code. Then execute the program and observe the behavior of the program.

Experiment 2.1

Currently, the program uses an array to store records but since there is no file to save records, for each execution, the phone book is initially empty. When the program is terminated, records are lost. To cope with these issues, modify the program in such a way that for each execution, records are read from an input file to the array. You should also save the array to the input file when the program is terminated. Delete operation of current code simply replaces related data with a blank. You should also eliminate these blanks while writing the array to the file.

Experiment 2.2

Add method of the current code adds new records to the end of the array. In this experiment, you are supposed to modify this method so that it adds new records in alphabetical order. You should also use pointer arithmetic to access the elements of the array, not index representation.

kayit.h

```
#define AD_UZUNLUK 30
#define TELNO_UZUNLUK 15

struct Tel_Kayit{
       char ad[AD_UZUNLUK];
    char telno[TELNO_UZUNLUK];
};
```

diziislemleri.h

```
#ifndef DIZIISLEMLERI_H
#define DIZIISLEMLERI_H
#include <stdio.h>
#include "kayit.h"

#define MAXKAYIT 100
struct Dizi{
       char *dosyaadi;
```

```
        FILE *teldefteri;
        Tel_Kayit k[MAXKAYIT];
        void olustur();
        void ekle(Tel_Kayit *);
        int ara(char []);
    void sil(int kayitno);
    void guncelle(int kayitno, Tel_Kayit *);
        void listele();
        int kayitsayisi;
};
#endif
```

diziislemleri.cpp

```
#include <iostream>
#include <stdlib.h>
//#include <cstdlib.h>
#include <stdio.h>
#include <string.h>
#include <iomanip>
#include "diziislemleri.h"

using namespace std;

void Dizi::ekle(Tel_Kayit *ykptr){
        strcpy(k[kayitsayisi].ad,ykptr->ad);
    strcpy(k[kayitsayisi].telno,ykptr->telno);
        kayitsayisi++;
}

void Dizi::olustur(){
        kayitsayisi=0;
}

int Dizi::ara(char aranacak[]){
        int sayac=0;
        bool tumu=false;
        int bulunan,flag=0;
        if(strcmp(aranacak,"*")==0)
                tumu=true;
        for(int i=0;i<kayitsayisi;i++){
                    if(strcmp(k[i].ad,aranacak)==0)
                       {
                      cout<<i+1<<". "<<k[i].ad<<endl;
                         bulunan=i;
                         flag=1;
                       }
        }
        return flag;
}

void Dizi::guncelle(int kayitno, Tel_Kayit *ykptr){
        strcpy(k[kayitno-1].ad,ykptr->ad);
    strcpy(k[kayitno-1].telno,ykptr->telno);
}

void Dizi::sil(int kayitno){
        Tel_Kayit temp;
        strcpy(temp.ad,  "");
        strcpy(temp.telno, "");
```

```
    int indisno;
    indisno=kayitno-1;
       k[indisno] = temp;
}

void Dizi::listele(){
       for(int i=0;i<kayitsayisi;i++)
              cout<<i+1<<". "<<k[i].ad<<"    \t"<<k[i].telno<<endl;
}
```

tel_prog.cpp

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>
//#include <conio.h>
#include <ctype.h>
#include "diziislemleri.h"

using namespace std;
typedef Dizi Veriyapisi;
Veriyapisi defter;

void menu_yazdir();
bool islem_yap(char);
void kayit_ara();
void kayit_ekle();
void kayit_sil();
void kayit_guncelle();
void kayit_listele();

int main(){
       defter.olustur();
       bool bitir = false;
       char secim;
       while (!bitir) {
              menu_yazdir();
              cin >> secim;
              bitir = islem_yap(secim);
              }
       return EXIT_SUCCESS;
}

void menu_yazdir(){
       //system("cls"); //linux'ta ekran temizleme icin system("clear"); kullaniniz
       cout << endl << endl;
       cout << "Phone Book Application" << endl;
       cout << "Choose an operation" << endl;
       cout << "A: Search Record" << endl;
       cout << "E: Add Record" << endl;
       cout << "G: Update Record" << endl;
       cout << "S: Delete Record" << endl;
       cout << "L: List Records" << endl;
       cout << "C: Exit" << endl;
       cout << endl;
       cout << "Enter an option {A, E, G, S, C, L} : ";
}

bool islem_yap(char secim){
       bool sonlandir=false;
       switch (secim) {
              case 'A': case 'a':
```

```
                        kayit_ara();
                        break;
                case 'E': case 'e':
                        kayit_ekle();
                        break;
                case 'G': case 'g':
                        kayit_guncelle();
                        break;
                case 'S': case 's':
                        kayit_sil();
                        break;
                case 'C': case 'c':
                        cout << "Are you sure that you want to terminate the program?
(E/H):";
                        cin >> secim;
                        if(secim=='E' || secim=='e')
                                sonlandir=true;
                                break;
          case 'L': case 'l':
                        kayit_listele();
                        break;
                default:
                        cout << "Error: You have made an invalid choice" << endl;
                        cout << "Try again {A, E, G, S, B, C} :" ;
                        cin >> secim;
                        sonlandir = islem_yap(secim);
                        break;
        }
        return sonlandir;
}

void kayit_ara(){
        char ad[AD_UZUNLUK];
        cout << "Please enter the name of the person you want to search (press '*' for
listing all):" << endl;
        cin.ignore(1000, '\n');
        cin.getline(ad,AD_UZUNLUK);
        if(defter.ara(ad)==0){
                cout << "Record can not be found" << endl;
        }
        getchar();
};
void kayit_ekle(){
        Tel_Kayit yenikayit;
        cout << "Please enter the information of the person you want to save " << endl;
        cout << "Name : " ;
        cin.ignore(1000, '\n');
        cin.getline(yenikayit.ad,AD_UZUNLUK);
        cout << "Phone number :";
        cin >> setw(TELNO_UZUNLUK) >> yenikayit.telno;
        defter.ekle(&yenikayit);
        cout << "Record has been added" << endl;
        getchar();
};

void kayit_sil(){
        char ad[AD_UZUNLUK];
        int secim;
        cout << "Please enter the name of the person you want to delete (press '*' for
listing all):" << endl;
        cin.ignore(1000, '\n');
        cin.getline(ad,AD_UZUNLUK);
```

```
	int kisisayisi=defter.ara(ad);
	if(kisisayisi==0){
		cout << "Record can not be found" << endl;
	}
	else {
		if (kisisayisi==1){
			cout << "Record has been found." << endl;
			cout << "Please enter the index of the record if you want to
delete this contact (Press -1 to exit without deletion): " ;
		}
		else
			cout << "Please enter the index of the record that you want to
delete (Press -1 to exit without deletion): " ;
		cin >> secim;
		if(secim==-1) return;
		defter.sil(secim);
		cout << "Record has been deleted" <<endl;
	}
	getchar();
};

void kayit_guncelle(){
	char ad[AD_UZUNLUK];
	int secim;
	cout << "Please enter the name of the person you want to update (press '*' for
listing all):" << endl;
	cin.ignore(1000, '\n');
	cin.getline(ad,AD_UZUNLUK);
	int kisisayisi=defter.ara(ad);
	if(kisisayisi==0){
		cout << "Record can not be found" << endl;
	}
	else {
		if (kisisayisi==1){
			cout << "Record has been found." << endl;
			cout << "Please enter the index of the record if you want to
update this contact (Press -1 to exit without updating) " ;
		}
		else
			cout << "Please enter the index of the record that you want to
update (Press -1 to exit without updating): " ;
		cin >> secim;
		if(secim==-1) return;
		Tel_Kayit yenikayit;
		cout << "Please enter the up-to-date information" << endl;
		cout << "Name : " ;
		cin.ignore(1000, '\n');
		cin.getline(yenikayit.ad,AD_UZUNLUK);
		cout << "Phone number :";
		cin >> setw(TELNO_UZUNLUK) >> yenikayit.telno;
		defter.guncelle(secim,&yenikayit);
		cout << "Record has been updated successfully" <<endl;
	}
	getchar();
};
void kayit_listele(){
	defter.listele();
	getchar();
}
```