BLG 233E DATA STRUCTURES AND LABORATORY

EXPERIMENT 1 – COMPILATION IN C++

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.

Experiment 1.1

In this experiment, you are supposed to write the given code. There must be 2 header files namely “record.h” and “fileoperations.h”. Furthermore, two source code file should exist which are named as “fileoperations.cpp” and “phone_prog.cpp”.

record.h

```
#define AD_UZUNLUK 30
#define TELNO_UZUNLUK 15

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

fileoperations.h

```
#ifndef DOSYAISLEMLERI_H
#define DOSYAISLEMLERI_H
#include <stdio.h>
#include "record.h"

struct Dosya{
	char *dosyaadi;
	FILE *teldefteri;
	void olustur();
	void kapat();
	void ekle(Tel_Kayit *);
	int ara(char []);
	void sil(int kayitno);
	void guncelle(int kayitno, Tel_Kayit *);
};
#endif
```

fileoperations.cpp

```
#include "fileoperations.h"
#include <iostream>
#include <stdlib.h>
#include <string.h>

using namespace std;
```

```
void Dosya::ekle(Tel_Kayit *ykptr){
	fseek(teldefteri, 0, SEEK_END);
	fwrite(ykptr, sizeof(Tel_Kayit), 1, teldefteri);
}

void Dosya::olustur(){
	dosyaadi="teldefteri.txt";
	teldefteri = fopen( dosyaadi, "r+" );
	if(!teldefteri){
		if(!(teldefteri = fopen( dosyaadi, "w+" ))){
			cerr << "File can not be opened" << endl;
			exit(1);
		}
	}
}

void Dosya::kapat(){
	fclose(teldefteri);
}

int Dosya::ara(char aranacak[]){
	Tel_Kayit k;
	int sayac=0;
	bool tumu=false;
	int bulunan=0;
	if(strcmp(aranacak,"*")==0)
		tumu=true;
	fseek(teldefteri, 0, SEEK_SET);
	while(!feof(teldefteri)){
		sayac++;
		fread( &k, sizeof (Tel_Kayit), 1, teldefteri);
		if(feof(teldefteri)) break;

		if(!tumu && strncmp(k.ad, aranacak, strlen(aranacak))!=0)
			continue;
		cout << sayac << "." << k.ad << " " << k.telno << endl;
		bulunan++;
	}
	return bulunan;
}

void Dosya::guncelle(int kayitno, Tel_Kayit *ykptr){
	if(fseek(teldefteri, sizeof(Tel_Kayit)*(kayitno-1), SEEK_SET)==0)
		fwrite(ykptr, sizeof(Tel_Kayit), 1, teldefteri);
}

void Dosya::sil(int kayitno){
	Tel_Kayit boskayit={"",""};
	if(fseek(teldefteri, sizeof(Tel_Kayit)*(kayitno-1), SEEK_SET)==0)
		fwrite(&boskayit, sizeof(Tel_Kayit), 1, teldefteri);
}
```

phone_prog.cpp

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>
#include <ctype.h>

#include "fileoperations.h"

using namespace std;

typedef Dosya Veriyapisi;
```

```
Veriyapisi defter;

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

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

void menu_yazdir(){
	system("clear");
	cout << endl << endl;
	cout << "Phone Book Application" << endl;
	cout << "Choose an option" << endl;
	cout << "A: Search Record" << endl;
	cout << "E: Add Record" << endl;
	cout << "G: Update Record" << endl;
	cout << "S: Delete Record" << endl;
	cout << "C: Exit" << endl;
	cout << endl;
	cout << "Enter your option {A, E, G, S, C} : ";
}

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;
		default:
			cout << "Error: You have made an invalid choice" << endl;
			cout << "Try again {A, E, G, S, 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();
};
```

Experiment 1.2

Compile the code that you have written, in both Visual Studio and SSH using g++ and create an executable.

Experiment 1.3

Observe the behavior of the program for each option and test your abilities of debugging the variables.

Experiment 1.4

In the deletion operation, deleted contact is replaced with a blank record. Implement a maintenance operation for eliminating these blank records. To do so, you should create a new temporary record file and move the records to this temporary file with ignoring blank records. At the end, copy the temporary file's content to the main file.