```cpp
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <iomanip>
#include <string.h>
#include <stdlib.h>
#include <cstdlib>
#include "fileoperations.h"

using namespace std;

void Operations::create(){
    numberofrecords = 0;
    phonebook = new Phone_Record[SIZE];
    size = SIZE;
}

void Operations::add(Phone_Record *newrecordptr){

    if (numberofrecords == size){
        Phone_Record * tempptr;
        tempptr = new Phone_Record[2*size];
        for (int i = 0; i < size; i++){
            tempptr[i].name = new char[strlen(phonebook[i].name)+1];
            tempptr[i].telno = new char[strlen(phonebook[i].telno)+1];
            strcpy(tempptr[i].name, phonebook[i].name);
            strcpy(tempptr[i].telno, phonebook[i].telno);
        }

        delete[] phonebook;
        size *= 2;
        phonebook = new Phone_Record[size];
        for (int i = 0; i < numberofrecords; i++){
            phonebook[i].name = new char[strlen(tempptr[i].name) + 1];
            phonebook[i].telno = new char[strlen(tempptr[i].telno) + 1];
            strcpy(phonebook[i].name, tempptr[i].name);
            strcpy(phonebook[i].telno, tempptr[i].telno);
            delete[] tempptr[i].name;
            delete[] tempptr[i].telno;
        }
        delete[] tempptr;
    }
    Phone_Record * ptr;
    ptr = &phonebook[numberofrecords];
    if (numberofrecords == 0) {

        ptr->name = new char[strlen(newrecordptr->name)+1];
        ptr->telno = new char[strlen(newrecordptr->telno)+1];
        strcpy(ptr->name, newrecordptr->name);
        strcpy(ptr->telno, newrecordptr->telno);
        numberofrecords++;
    }
    else {
        int i = 0;
        while (i < numberofrecords && strcmp(newrecordptr->name,phonebook[i].name ) >0 )
            i++;
            int temp = numberofrecords;
            for (int j = 0; j < numberofrecords - i - 1 ;j++){
                phonebook[temp].name = new char[strlen(phonebook[temp - 1].name)+1];
                phonebook[temp].telno = new char[strlen(phonebook[temp - 1].telno)+1];
                strcpy(phonebook[temp].name, phonebook[temp - 1].name);
                strcpy(phonebook[temp].telno, phonebook[temp - 1].telno);
                temp--;
            }
            ptr = &phonebook[i];

            ptr->name = new char[strlen(newrecordptr->name)+1];
            ptr->telno = new char[strlen(newrecordptr->telno)+1];
            strcpy(ptr->name, newrecordptr->name);
            strcpy(ptr->telno, newrecordptr->telno);
            numberofrecords++;
    }


}

void Operations::list(){
    for (int i = 0; i < numberofrecords; i++){
        cout << i + 1 << ". " << phonebook[i].name << " " << phonebook[i].telno << endl;
    }
    getchar();
    getchar();

}

void Operations::close(){
    delete[] phonebook;
}

void Operations::del(int indexno){
    int temp=0;
    for (int i = indexno - 1; i < numberofrecords - 2; i++){
        delete phonebook[i].name;
        delete phonebook[i].telno;
        phonebook[i].name = new char[strlen(phonebook[i + 1].name)+1];
        strcpy(phonebook[i].name, phonebook[i + 1].name);
        phonebook[i].telno = new char[strlen(phonebook[i + 1].telno) + 1];
        strcpy(phonebook[i].telno, phonebook[i + 1].telno);
        int temp = i;
    }


    numberofrecords--;
```