

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <iostream>
3  #include <iomanip>
4  #include <string.h>
5  #include <stdlib.h>
6  #include <cstdlib>
7  #include "fileoperations.h"
8
9  using namespace std;
10
11 void Operations::create(){
12     numberofrecords = 0;
13     phonebook = new Phone_Record[SIZE];
14     size = SIZE;
15 }
16
17 void Operations::add(Phone_Record *newrecordptr){
18
19     if (numberofrecords == size){
20         Phone_Record * tempptr;
21         tempptr = new Phone_Record[2*size];
22         for (int i = 0; i < size; i++){
23             tempptr[i].name = new char[strlen(phonebook[i].name)+1];
24             tempptr[i].telno = new char[strlen(phonebook[i].telno)+1];
25             strcpy(tempptr[i].name, phonebook[i].name);
26             strcpy(tempptr[i].telno, phonebook[i].telno);
27         }
28
29         delete[] phonebook;
30         size *= 2;
31         phonebook = new Phone_Record[size];
32         for (int i = 0; i < numberofrecords; i++){
33             phonebook[i].name = new char[strlen(tempptr[i].name) + 1];
34             phonebook[i].telno = new char[strlen(tempptr[i].telno) + 1];
35             strcpy(phonebook[i].name, tempptr[i].name);
36             strcpy(phonebook[i].telno, tempptr[i].telno);
37             delete[] tempptr[i].name;
38             delete[] tempptr[i].telno;
39         }
40         delete[] tempptr;
41     }
42     Phone_Record * ptr;
43     ptr = &phonebook[numberofrecords];
44     if (numberofrecords == 0) {
45
46         ptr->name = new char[strlen(newrecordptr->name)+1];
47         ptr->telno = new char[strlen(newrecordptr->telno)+1];
48         strcpy(ptr->name, newrecordptr->name);
49         strcpy(ptr->telno, newrecordptr->telno);
50         numberofrecords++;
51     }
52     else {
53         int i = 0;
54         while (i < numberofrecords && strcmp(newrecordptr->name, phonebook[i].name) > 0 )
55             i++;
56         int temp = numberofrecords;
57         for (int j = 0; j < numberofrecords - i - 1 ; j++){
58             phonebook[temp].name = new char[strlen(phonebook[temp - 1].name)+1];
59             phonebook[temp].telno = new char[strlen(phonebook[temp - 1].telno)+1];
60             strcpy(phonebook[temp].name, phonebook[temp - 1].name);
61             strcpy(phonebook[temp].telno, phonebook[temp - 1].telno);
62             temp--;
63         }
64         ptr = &phonebook[i];
65
66         ptr->name = new char[strlen(newrecordptr->name)+1];

```

```

67         ptr->telno = new char[strlen(newrecordptr->telno)+1];
68         strcpy(ptr->name, newrecordptr->name);
69         strcpy(ptr->telno, newrecordptr->telno);
70         numberofrecords++;
71     }
72
73
74 }
75
76 void Operations::list(){
77     for (int i = 0; i < numberofrecords; i++){
78         cout << i + 1 << ". " << phonebook[i].name << " " << phonebook[i].telno << endl;
79     }
80     getchar();
81     getchar();
82
83 }
84
85 void Operations::close(){
86     delete[] phonebook;
87 }
88
89 void Operations::del(int indexno){
90     int temp=0;
91     for (int i = indexno - 1; i < numberofrecords - 2; i++){
92         delete phonebook[i].name;
93         delete phonebook[i].telno;
94         phonebook[i].name = new char[strlen(phonebook[i + 1].name)+1];
95         strcpy(phonebook[i].name, phonebook[i + 1].name);
96         phonebook[i].telno = new char[strlen(phonebook[i + 1].telno) + 1];
97         strcpy(phonebook[i].telno, phonebook[i + 1].telno);
98         int temp = i;
99     }
100
101
102     numberofrecords--;
103
104

```