

1BI18IS067

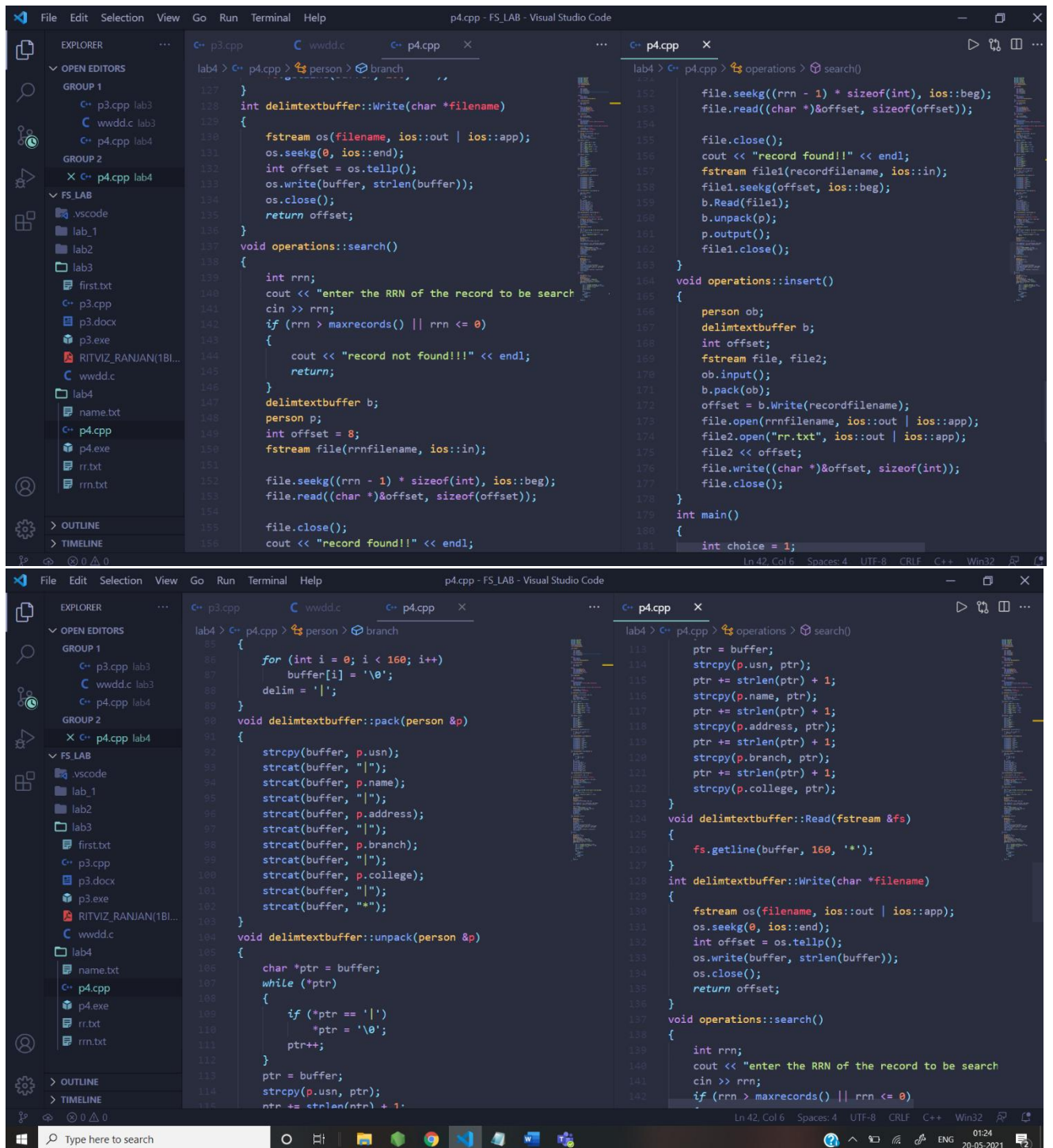
FS LAB-4

RITVIZ RANJAN

18/05/21

```
lab4 > C++ p3.cpp > person > branch
1 #include <iostream>
2 #include <stdio.h>
3 #include <fstream>
4 #include <stdlib.h>
5 #include <string.h>
6 using namespace std;
7 class delimtextbuffer;
8
9 class person
10 {
11     char usn[20];
12     char name[18];
13     char address[20];
14     char branch[20];
15     char college[20];
16
17 public:
18     void input();
19     void output();
20     friend class delimtextbuffer;
21 };
22 class delimtextbuffer
23 {
24     char buffer[160];
25     char delim;
26
27 public:
28     void pack(person &p);
29     void unpack(person &p);
30     void Read(fstream &fs);
31 };
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
lab4 > C++ p4.cpp > person > branch
33 };
34 class operations
35 {
36     char *rrnfilename;
37     char *recordfilename;
38
39 public:
40     int maxrecords();
41     operations(char *rrnfile, char *recordfile);
42     void search();
43     void insert();
44 };
45 operations::operations(char *rrnfile, char *recordfil
46 {
47     rrnfilename = rrnfile;
48     recordfilename = recordfile;
49 }
50 int operations::maxrecords()
51 {
52     fstream file(rrnfilename, ios::in);
53     int pos;
54     file.seekg(0, ios::end);
55     pos = file.tellg();
56     return (pos / (sizeof(int)));
57 }
58 void person::input()
59 {
60     cout << "Enter usn" << endl;
61     cin >> usn;
62     cout << "Enter name" << endl;
```



The screenshot shows the Visual Studio Code interface with the file explorer on the left and two editor windows open. The Explorer shows a project named 'FS_LAB' with subfolders 'lab1', 'lab2', 'lab3', and 'lab4'. The 'lab4' folder contains files like 'name.txt', 'p4.cpp', 'p4.exe', 'rr.txt', and 'rrn.txt'. The 'p4.cpp' file is open in two editor windows. The left window shows the implementation of the `delimittextbuffer` class, including the `delimittextbuffer()` constructor, a `for` loop to initialize a buffer, and the `pack` and `unpack` methods. The `pack` method concatenates fields like `usn`, `name`, `address`, `branch`, and `college` into a buffer. The `unpack` method reads the buffer back into a `person` object. The right window shows the `Read` and `Write` methods, which use `fstream` to read from and write to a file. The `Write` method also includes a `search` method.

```
lab4 > C++ p4.cpp > person > branch
84 delimittextbuffer::delimittextbuffer()
85 {
86     for (int i = 0; i < 160; i++)
87         buffer[i] = '\0';
88     delim = '|';
89 }
90 void delimittextbuffer::pack(person &p)
91 {
92     strcpy(buffer, p.usn);
93     strcat(buffer, "|");
94     strcat(buffer, p.name);
95     strcat(buffer, "|");
96     strcat(buffer, p.address);
97     strcat(buffer, "|");
98     strcat(buffer, p.branch);
99     strcat(buffer, "|");
100    strcat(buffer, p.college);
101    strcat(buffer, "|");
102    strcat(buffer, "*");
103 }
104 void delimittextbuffer::unpack(person &p)
105 {
106     char *ptr = buffer;
107     while (*ptr)
108     {
109         if (*ptr == '|')
110             *ptr = '\0';
111         ptr++;
112     }
113     ptr = buffer;
```

```
lab4 > C++ p4.cpp > person > branch
109     if (*ptr == '|')
110         *ptr = '\0';
111     ptr++;
112 }
113 ptr = buffer;
114 strcpy(p.usn, ptr);
115 ptr += strlen(ptr) + 1;
116 strcpy(p.name, ptr);
117 ptr += strlen(ptr) + 1;
118 strcpy(p.address, ptr);
119 ptr += strlen(ptr) + 1;
120 strcpy(p.branch, ptr);
121 ptr += strlen(ptr) + 1;
122 strcpy(p.college, ptr);
123 }
124 void delimittextbuffer::Read(fstream &fs)
125 {
126     fs.getline(buffer, 160, '*');
127 }
128 int delimittextbuffer::Write(char *filename)
129 {
130     fstream os(filename, ios::out | ios::app);
131     os.seekg(0, ios::end);
132     int offset = os.tellp();
133     os.write(buffer, strlen(buffer));
134     os.close();
135     return offset;
136 }
137 void operations::search()
138 {
139     int crr;
```

The screenshot shows the Visual Studio Code interface with the file explorer on the left and two editor windows open. The Explorer shows the same project structure as the first screenshot. The 'p4.cpp' file is open in two editor windows. The left window shows the implementation of the `person` class, including the `input` and `output` methods. The `input` method prompts the user to enter `usn`, `name`, `address`, `branch`, and `college`. The `output` method displays the stored information. The `delimittextbuffer` class is also shown, including the `delimittextbuffer()` constructor, the `pack` method, and the `unpack` method. The `pack` method concatenates the fields of a `person` object into a buffer. The `unpack` method reads the buffer back into a `person` object. The right window shows the `Read` and `Write` methods, which use `fstream` to read from and write to a file. The `Write` method also includes a `search` method.

```
lab4 > C++ p4.cpp > person > branch
56     return (pos / (sizeof(int)));
57 }
58 void person::input()
59 {
60     cout << "Enter usn" << endl;
61     cin >> usn;
62     cout << "Enter name" << endl;
63     cin >> name;
64     cout << "Enter address" << endl;
65     cin >> address;
66     cout << "Enter branch" << endl;
67     cin >> branch;
68     cout << "Enter college" << endl;
69     cin >> college;
70 }
71 void person::output()
72 {
73     cout << "USN:";
74     puts(usn);
75     cout << "Name:";
76     puts(name);
77     cout << "Address:";
78     puts(address);
79     cout << "Branch:";
80     puts(branch);
81     cout << "College:";
82     puts(college);
83 }
84 delimittextbuffer::delimittextbuffer()
85 {
86     for (int i = 0; i < 160; i++)
```

```
lab4 > C++ p4.cpp > operations > search()
84 delimittextbuffer::delimittextbuffer()
85 {
86     for (int i = 0; i < 160; i++)
87         buffer[i] = '\0';
88     delim = '|';
89 }
90 void delimittextbuffer::pack(person &p)
91 {
92     strcpy(buffer, p.usn);
93     strcat(buffer, "|");
94     strcat(buffer, p.name);
95     strcat(buffer, "|");
96     strcat(buffer, p.address);
97     strcat(buffer, "|");
98     strcat(buffer, p.branch);
99     strcat(buffer, "|");
100    strcat(buffer, p.college);
101    strcat(buffer, "|");
102    strcat(buffer, "*");
103 }
104 void delimittextbuffer::unpack(person &p)
105 {
106     char *ptr = buffer;
107     while (*ptr)
108     {
109         if (*ptr == '|')
110             *ptr = '\0';
111         ptr++;
112     }
```

```
lab4 > C++ p4.cpp > person > branch
157 fstream file1(recordfilename, ios::in);
158 file1.seekg(offset, ios::beg);
159 b.Read(file1);
160 b.unpack(p);
161 p.output();
162 file1.close();
163 }
164 void operations::insert()
165 {
166     person ob;
167     delimittextbuffer b;
168     int offset;
169     fstream file, file2;
170     ob.input();
171     b.pack(ob);
172     offset = b.Write(recordfilename);
173     file.open(rrnfilename, ios::out | ios::app);
174     file2.open("rr.txt", ios::out | ios::app);
175     file2 << offset;
176     file.write((char *)&offset, sizeof(int));
177     file.close();
178 }
179 int main()
180 {
181     int choice = 1;
182     fstream file, file1;
183     person ob;
184     delimittextbuffer b;
185     char filename[20] = "name.txt";
186     char rrnfilename[20] = "rrn.txt";
187     operations o(rrnfilename, filename);
188     while (choice < 3)
189     {
190         cout << "1:Insert a Record" << endl;
191         cout << "2:Search for a Record" << endl;
192         cin >> choice;
193         switch (choice)
194         {
195             case 1:
196                 o.insert();
197                 break;
198             case 2:
199                 o.search();
200                 break;
201         }
202     }
203     return 0;
204 }
```

```
Enter name
roshan
Enter address
mumbai
Enter branch
cse
Enter college
dsce
1:Insert a Record
2:Search for a Record
3:exit
1
Enter usn
103
Enter name
yash
Enter address
patna
Enter branch
ece
Enter college
bms
1:Insert a Record
2:Search for a Record
3:exit
2
enter the RRN of the record to be searched(1 - based)3
record found!!
USN:103
Name:yash
Address:patna
Branch:ece
College:bms
1:Insert a Record
2:Search for a Record
3:exit
2
```

The screenshot shows the Visual Studio Code interface with a C++ file named `p4.cpp` open. The file contains a program that manages a list of records. The terminal output shows the program's execution, including prompts for record insertion and searching, and the resulting data structure.

```
College:bms
1:Insert a Record
2:Search for a Record
3:exit
2
enter the RRN of the record to be searched(1 - based)2
record found!!
USN:iz
Name:surat
Address:ise
Branch:bit
College:
1:Insert a Record
2:Search for a Record
3:exit
2
enter the RRN of the record to be searched(1 - based)1
record found!!
USN:101
Name:ritviz
Address:surat
Branch:ise
College:bit
1:Insert a Record
2:Search for a Record
3:exit
```

The screenshot shows a Notepad window titled `rrn.txt - Notepad`. The text content of the file is:

```
0 6 N h „
```

The screenshot shows a Notepad window titled `rr.txt - Notepad`. The text content of the file is:

```
0265478104132
```

The screenshot shows a Notepad window titled `name.txt - Notepad`. The text content of the file is:

```
101|ritviz|surat|ise|bit|*102|roshan|mumbai|cse|dsce|*103|yash|patna|ece|bms|*101|ritviz|surat|ise|bit|*102|roshan|mumbai|cse|dsce|*103|yash|patna|ece|bms|*
```