

**Problem 01:**

```
Codes Lab-11_24k-2000 > C Problem_01.c > main()
1  #include <stdio.h>
2  #include <string.h>
3  // Structure to store information about students, including
4  // academic details, and address and a program to input de
5  // and display the student with the highest GPA.
6  struct Address
7  {
8      char city[50];
9      char street[50];
10     int postalCode;
11 };
12
13 struct Student
14 {
15     char name[50];
16     int age;
17     int rollNumber;
18     float gpa;
19     struct Address address;
20 };
21
22 int findHighestGPA(struct Student students[], int n)
23 {
24     int index = 0;
25     for (int i = 1; i < n; i++)
26     {
27         if (students[i].gpa > students[index].gpa)
28         {
29             index = i;
30         }
31     }
32     return index;
33 }
34
35 int main()
36 {
37     struct Student students[5];
38     int i;
39
40     printf("Enter details for 5 students:\n");
```

Student 1:  
Name: ali  
Age: 18  
Roll Number: 2049  
GPA: 3.2  
City: karachi  
Street: khayabaneyItehaad  
Postal Code: 0412

Student 2:  
Name: qasim  
Age: 17  
Roll Number: 2099  
GPA: 2.9  
City: karachi  
Street: Street-24  
Postal Code: 0410

Student 3:  
Name: ahsan  
Age: 16  
Roll Number: 2039  
GPA: 3.1  
City: karachi  
Street: road-9  
Postal Code: 0121

Student 4:  
Name: ayan  
Age: 18  
Roll Number: 2021  
GPA: 3.15  
City: karachi  
Street: road-69  
Postal Code: 0413

Student 5:  
Name: zara  
Age: 18  
Roll Number: 2000  
GPA: 3.12  
City: karachi  
Street: loopRoad  
Postal Code: 1042

Codes Lab-11\_24k-2000 &gt; C Problem\_01.c &gt; main()

```

35  int main()
40  printf("Enter details for 5 students:\n");
41  for (i = 0; i < 5; i++)
42  {
43      printf("\nStudent %d:\n", i + 1);
44      printf("Name: ");
45      scanf("%s", students[i].name);
46      printf("Age: ");
47      scanf("%d", &students[i].age);
48      printf("Roll Number: ");
49      scanf("%d", &students[i].rollNumber);
50      printf("GPA: ");
51      scanf("%f", &students[i].gpa);
52      printf("City: ");
53      scanf("%s", students[i].address.city);
54      printf("Street: ");
55      scanf("%s", students[i].address.street);
56      printf("Postal Code: ");
57      scanf("%d", &students[i].address.postalCode);
58  }
59
60  int highestIndex = findHighestGPA(students, 5);
61  printf("\nStudent with the highest GPA:\n");
62  printf("Name: %s\n", students[highestIndex].name);
63  printf("Age: %d\n", students[highestIndex].age);
64  printf("Roll Number: %d\n", students[highestIndex].rollNumber);
65  printf("GPA: %.2f\n", students[highestIndex].gpa);
66  printf("Address: %s, %s, %d\n",
67      students[highestIndex].address.city,
68      students[highestIndex].address.street,
69      students[highestIndex].address.postalCode);
70
71  return 0;
72  }
73

```

Age: 17  
Roll Number: 2099  
GPA: 2.9  
City: karachi  
Street: Street-24  
Postal Code: 0410

Student 3:  
Name: ahsan  
Age: 16  
Roll Number: 2039  
GPA: 3.1  
City: karachi  
Street: road-9  
Postal Code: 0121

Student 4:  
Name: ayan  
Age: 18  
Roll Number: 2021  
GPA: 3.15  
City: karachi  
Street: road-69  
Postal Code: 0413

Student 5:  
Name: zara  
Age: 18  
Roll Number: 2000  
GPA: 3.12  
City: karachi  
Street: loopRoad  
Postal Code: 1042

Student with the highest GPA:  
Name: ali  
Age: 18  
Roll Number: 2049  
GPA: 3.20  
Address: karachi, khayabaneyItehaad, 412

**Problem 02:**

Codes Lab-11\_24k-2000 &gt; C Problem\_02.c &gt; ...

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  // program to manage employee records in a company. Store employee details in
5  // an array of structures and implement the following:
6  // Add new employees.
7  // Search for an employee by ID.
8  // Update salary details for a specific employee.
9  // Display all employee details sorted by their joining date
10
11 #define MAX_EMPLOYEES 10
12
13 struct employee {
14     char employee_id[10];
15     char name[20];
16     char role[15];
17     float salary;
18     char dept[10];
19     char join_date[15];
20 };
21
22
23 struct employee e[MAX_EMPLOYEES];
24 int employee_count = 0;
25 void add_new_employees() {
26     if (employee_count >= MAX_EMPLOYEES) {
27         printf("Employee limit reached. Cannot add more employees.\n");
28         return;
29     }
30
31     printf("==== Input Data for Employee %d ==== \n", employee_count + 1);
32     printf("Employee ID: ");
33     scanf("%s", e[employee_count].employee_id);
34     printf("Employee Name: ");
35     scanf("%s", e[employee_count].name);
36     printf("Employee Role: ");
37     scanf("%s", e[employee_count].role);
38     printf("Employee Salary: ");
39     scanf("%f", &e[employee_count].salary);
40     printf("Employee Department: ");
41     scanf("%s", e[employee_count].dept);
42     printf("Joining Date (DD/MM/YYYY): ");
43     scanf("%s", e[employee_count].join_date);
44
45     employee_count++;
46     printf("Employee added successfully....\n");
47 }
48
49 void search_employee() {

```

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 1

==== Input Data for Employee 1 ====

Employee ID: 2010

Employee Name: Ahsan

Employee Role: Assistant

Employee Salary: 40000

Employee Department: IT

Joining Date (DD/MM/YYYY): 21/12/2012

Employee added successfully....

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 2

Enter the ID of the employee you want to search for: 2010

Employee Found:

==== EMPLOYEE DETAILS ====

ID: 2010

Name: Ahsan

Role: Assistant

Department: IT

Salary: 40000.00

Joining Date: 21/12/2012

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 3

Enter the ID of the employee whose salary you want to update: 2010

Employee Found:

Current Salary: 40000.00

Enter the new salary: 45000

Salary updated successfully!

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Codes Lab-11\_24k-2000 &gt; C Problem\_02.c &gt; ...

```

49 void search_employee() {
50     char id_search[10];
51     printf("Enter the ID of the employee you want to search for: ");
52     scanf("%s", id_search);
53
54     for (int i = 0; i < employee_count; i++) {
55         if (strcmp(e[i].employee_id, id_search) == 0) {
56             printf("Employee Found:\n");
57             printf("==== EMPLOYEE DETAILS ==== \n");
58             printf("ID: %s\n", e[i].employee_id);
59             printf("Name: %s\n", e[i].name);
60             printf("Role: %s\n", e[i].role);
61             printf("Department: %s\n", e[i].dept);
62             printf("Salary: %.2f\n", e[i].salary);
63             printf("Joining Date: %s\n", e[i].join_date);
64             return;
65         }
66     }
67     printf("No employee found with this ID.\n");
68 }
69
70 void update_salary() {
71     char id_search[10];
72     float new_salary;
73     printf("Enter the ID of the employee whose salary you want to update: ");
74     scanf("%s", id_search);
75
76     for (int i = 0; i < employee_count; i++) {
77         if (strcmp(e[i].employee_id, id_search) == 0) {
78             printf("Employee Found:\n");
79             printf("Current Salary: %.2f\n", e[i].salary);
80             printf("Enter the new salary: ");
81             scanf("%f", &new_salary);
82             e[i].salary = new_salary;
83             printf("Salary updated successfully!\n");
84             return;
85         }
86     }
87     printf("No employee found with this ID.\n");
88 }
89
90 void display_all_details() {
91     if (employee_count == 0) {
92         printf("No employees to display.\n");
93         return;
94     }
95
96     printf("\n%-10s %-20s %-15s %-10s %-10s %-15s\n", "ID", "Name", "Role", "Sa");
97     printf("-----");

```

Enter your choice: 4

ID	Name	Role	Salary	Dept	Join Date
2010	Ahsan	Assistant	45000.00	IT	21/12/2012

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 1

==== Input Data for Employee 2 ====

Employee ID: 2011

Employee Name: Alina

Employee Role: admin

Employee Salary: 80000

Employee Department: reception

Joining Date (DD/MM/YYYY): 12/12/2016

Employee added successfully....

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 4

ID	Name	Role	Salary	Dept	Join Date
2010	Ahsan	Assistant	45000.00	IT	21/12/2012
2011	Alina	admin	80000.00	reception	12/12/2016

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 5

Employees sorted by joining date.

1. Add New Employees
2. Search for an Employee by ID
3. Update Salary Details
4. Display All Employee Details
5. Sort Employees by Joining Date
6. Exit

Enter your choice: 6

Exiting the program. Goodbye!

```
Codes Lab-11_24k-2000 > C Problem_02.c > main()
90 void display_all_details() {
96     printf("\n%-10s %-20s %-15s %-10s %-10s %-15s\n", "ID", "Name", "Role", "Salary", "Dept", "Join Date");
97     printf("-----\n");
98     for (int i = 0; i < employee_count; i++) {
99         printf("%-10s %-20s %-15s %-10.2f %-10s %-15s\n", e[i].employee_id, e[i].name, e[i].role, e[i].salary, e[i].dept, e[i].join_date);
100     }
101 }
102 void sort_the_employees() {
103     struct employee temp;
104     for (int i = 0; i < employee_count - 1; i++) {
105         for (int j = 0; j < employee_count - i - 1; j++) {
106             if (strcmp(e[j].join_date, e[j + 1].join_date) > 0) {
107                 temp = e[j];
108                 e[j] = e[j + 1];
109                 e[j + 1] = temp;
110             }
111         }
112     }
113     printf("Employees sorted by joining date.\n");
114 }
115 int main() {
116     int choice;
117     while (1) {
118         printf("\n1. Add New Employees\n2. Search for an Employee by ID\n3. Update Salary Details\n4. Display All Employee Details\n5. Sort Employees by Joining Date\n6. Exit\n");
119         printf("Enter your choice: ");
120         scanf("%d", &choice);
121
122         switch (choice) {
123             case 1:
124                 add_new_employees();
125                 break;
126             case 2:
127                 search_employee();
128                 break;
129             case 3:
130                 update_salary();
131                 break;
132             case 4:
133                 display_all_details();
134                 break;
135             case 5:
136                 sort_the_employees();
137                 break;
138             case 6:
139                 printf("Exiting the program. Goodbye!\n");
140                 exit(0);
141             default:
142                 printf("Invalid choice. Please try again.\n");
143         }
144     }
145     return 0;
}
```



**Problem 03:**

Codes Lab-11\_24k-2000 &gt; C Problem\_03.c &gt; ...

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  // Develop a grading system using text files. Program to input and store
5  // names, roll numbers, and grades into a text file. Allow the user to v
6  // a grade above a certain threshold.
7
8  const char *students_data = "students.txt";
9
10 struct students {
11     char student_name[15];
12     char roll_no[10];
13     float grade;
14 };
15
16 void add_data() {
17     struct students s;
18
19     printf("==== INPUT STUDENT DATA ==== \n");
20     printf("Student Name: ");
21     scanf("%s", s.student_name);
22     printf("Student Roll No: ");
23     scanf("%s", s.roll_no);
24     printf("Student Grades: ");
25     scanf("%f", &s.grade);
26
27     FILE *file = fopen(students_data, "a");
28     if (file == NULL) {
29         printf("Error: Unable to open file for writing. \n");
30         return;
31     }
32
33     fprintf(file, "%s %s %.2f \n", s.student_name, s.roll_no, s.grade);
34     fclose(file);
35
36     printf("Student data added successfully! \n");
37 }
38
39 void view_records() {
40     char search_roll[10];
41     struct students s[100];
42     int s_count = 0;
43     int flag = 0;
44
45     printf("Enter the roll no of the student you want to view: ");
46     scanf("%s", search_roll);
47
48     FILE *file = fopen(students_data, "r");
49     if (file == NULL) {

```

```

===== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
===== INPUT STUDENT DATA =====
Student Name: zara
Student Roll No: 2101
Student Grades: 9
Student data added successfully!
===== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
===== INPUT STUDENT DATA =====
Student Name: Ali
Student Roll No: 2102
Student Grades: 9
Student data added successfully!
===== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
===== INPUT STUDENT DATA =====
Student Name: Ahsan
Student Roll No: 2310
Student Grades: 10
Student data added successfully!
===== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
===== INPUT STUDENT DATA =====
Student Name: Qasim
Student Roll No: 1201
Student Grades: 4
Student data added successfully!
===== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 2
Enter the roll no of the student you want to view: 2101

>>>> Student Record Found!!!
--- Student Details ---
Student Name: zara
Student Roll No: 2101
Student Grades: 9.00
===== Students Menu =====
1. Add Student Data

```

```

39 void view_records() {
48     FILE *file = fopen(students_data, "r");
49     if (file == NULL) {
50         printf("The file does not exist...\n");
51         return;
52     }
53
54     while (fscanf(file, "%s %s %f", s[s_count].student_name, s[s_count].roll_no, &s[s_count].grade) == 3){
55         s_count++;
56     }
57     fclose(file);
58
59     for (int i = 0; i < s_count; i++) {
60         if (strcmp(s[i].roll_no, search_roll) == 0) {
61             flag = 1;
62             printf("\n\n>>>> Student Record Found!!! \n");
63             printf("--- Student Details ---\n");
64             printf("Student Name: %s\n", s[i].student_name);
65             printf("Student Roll No: %s\n", s[i].roll_no);
66             printf("Student Grades: %.2f\n", s[i].grade);
67             break;
68         }
69     }
70
71     if (flag == 0) {
72         printf("No student record found with roll number: %s\n", search_roll);
73     }
74 }
75
76 int main() {
77     int choice = 0;
78
79     while (choice != 3) {
80         printf("==== Students Menu =====\n");
81         printf("1. Add Student Data\n");
82         printf("2. View Student Record\n");
83         printf("3. Exit the Program\n");
84         printf(">> ");
85         scanf("%d", &choice);
86
87         switch (choice) {
88             case 1:
89                 add_data();
90                 break;
91
92             case 2:
93                 view_records();
94                 break;
95
96             case 3:
97                 printf("Exiting the program. Goodbye!\n");
98                 return 0;
99         }
100     }
101 }

```

```

>> 1
==== INPUT STUDENT DATA ====
Student Name: Ali
Student Roll No: 2102
Student Grades: 9
Student data added successfully!
==== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
==== INPUT STUDENT DATA ====
Student Name: Ahsan
Student Roll No: 2310
Student Grades: 10
Student data added successfully!
==== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 1
==== INPUT STUDENT DATA ====
Student Name: Qasim
Student Roll No: 1201
Student Grades: 4
Student data added successfully!
==== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 2
Enter the roll no of the student you want to view: 2101

>>>> Student Record Found!!!
--- Student Details ---
Student Name: zara
Student Roll No: 2101
Student Grades: 9.00
==== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 2
Enter the roll no of the student you want to view: 2103
No student record found with roll number: 2103
==== Students Menu =====
1. Add Student Data
2. View Student Record
3. Exit the Program
>> 3
Exiting the program. Goodbye!
PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs
24k-2000>

```

Codes Lab-11\_24k-2000 &gt; C Problem\_03.c &gt; ...

```
76  int main() {
77      int choice = 0;
78
79      while (choice != 3) {
80          printf("===== Students Menu =====\n");
81          printf("1. Add Student Data\n");
82          printf("2. View Student Record\n");
83          printf("3. Exit the Program\n");
84          printf(">> ");
85          scanf("%d", &choice);
86
87          switch (choice) {
88              case 1:
89                  add_data();
90                  break;
91
92              case 2:
93                  view_records();
94                  break;
95
96              case 3:
97                  printf("Exiting the program. Goodbye!\n");
98                  break;
99
100             default:
101                 printf("Invalid Input. Please try again.\n");
102                 break;
103         }
104     }
105
106     return 0;
107 }
108
```

```
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```



**Problem 04:**

Codes Lab-11\_24k-2000 &gt; C Problem\_04.c &gt; view\_records()

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  // Write a program to store information about books in a li
5  // Implement the following operations:
6  // Add new books.
7  // Search for a book by title or author.
8  // Display all books sorted by year of publication.
9
10 const char *booksdb = "books.txt";
11
12 struct library {
13     char title[30];
14     char author[20];
15     char genre[15];
16     char pub_date[15];
17 };
18
19 struct library b[100];
20 int b_count = 0;
21
22 void add_data() {
23     printf("==== INPUT BOOK DATA ==== \n");
24     printf("Book Title: ");
25     scanf(" %s", b[b_count].title);
26     printf("Book Author: ");
27     scanf(" %s", b[b_count].author);
28     printf("Book Genre: ");
29     scanf(" %s", b[b_count].genre);
30     printf("Publishing Date (YYYY-MM-DD): ");
31     scanf(" %s", b[b_count].pub_date);
32
33     FILE *file = fopen(booksdb, "a");
34     if (file == NULL) {
35         printf("Can't open file.... Try again... \n");
36         return;
37     }
38
39     fwrite(&b[b_count], sizeof(struct library), 1, file);
40
41     fclose(file);
42     printf("Book data added successfully! \n");
43     b_count++;
44 }
45

```

```

===== Library Menu =====

```

1. Add Book Data
  2. Search for a Book
  3. Display All Books
  4. Exit the Program
- >> 1

```

===== INPUT BOOK DATA =====

```

```

Book Title: HarryPotter
Book Author: Harvidaton
Book Genre: adventure
Publishing Date (YYYY-MM-DD): 1999-12-12
Book data added successfully!

```

```

===== Library Menu =====

```

1. Add Book Data
  2. Search for a Book
  3. Display All Books
  4. Exit the Program
- >> 1

```

===== INPUT BOOK DATA =====

```

```

Book Title: CALCULUS19
Book Author: Alama
Book Genre: educational
Publishing Date (YYYY-MM-DD): 2021-12
Book data added successfully!

```

```

===== Library Menu =====

```

1. Add Book Data
  2. Search for a Book
  3. Display All Books
  4. Exit the Program
- >> 3

```

Books sorted by publication date.

```

```

===== LIST OF BOOKS =====

```

```

--- Book 1 ---

```

```

Title: HarryPotter
Author: Harvidaton
Genre: adventure
Publication Date: 1999-12-12

```

```

--- Book 2 ---

```

```

Title: CALCULUS19
Author: Alama
Genre: educational
Publication Date: 2021-12

```

```

===== Library Menu =====

```

1. Add Book Data
2. Search for a Book

```

46 void sort_books() {
47     struct library temp;
48     for (int i = 0; i < b_count - 1; i++) {
49         for (int j = 0; j < b_count - i - 1; j++) {
50             if (strcmp(b[j].pub_date, b[j + 1].pub_date) > 0) {
51                 temp = b[j];
52                 b[j] = b[j + 1];
53                 b[j + 1] = temp;
54             }
55         }
56     }
57     printf("Books sorted by publication date.\n");
58 }
59
60 void view_records() {
61     char search[30];
62     int flag = 0;
63
64     printf("Enter the title or author of the book you want to search: ");
65     scanf("%s", search);
66
67     FILE *file = fopen("booksdb", "r");
68     if (file == NULL) {
69         printf("The file does not exist...\n");
70         return;
71     }
72
73     while (fread(&b[b_count], sizeof(struct library), 1, file)) {
74         b_count++;
75     }
76     fclose(file);
77
78     for (int i = 0; i < b_count; i++) {
79         if (strcmp(b[i].title, search) == 0 || strcmp(b[i].author, search) == 0) {
80             flag = 1;
81             printf("\n\n>>>> Book Record Found!!! \n");
82             printf("--- Book Details ---\n");
83             printf("Book Title: %s\n", b[i].title);
84             printf("Book Author: %s\n", b[i].author);
85             printf("Book Genre: %s\n", b[i].genre);
86             printf("Publication Date: %s\n", b[i].pub_date);
87             break;
88         }
89     }
90
91     if (!flag) {
92         printf("No book record found with the search: %s\n", search);
93     }
94 }

```

```

1. Add Book Data
2. Search for a Book
3. Display All Books
4. Exit the Program
>> 3
Books sorted by publication date.
==== LIST OF BOOKS ====

```

```

--- Book 1 ---
Title: HarryPotter
Author: Harvidaton
Genre: adventure
Publication Date: 1999-12-12

```

```

--- Book 2 ---
Title: CALCULUS19
Author: Alama
Genre: educational
Publication Date: 2021-12

```

```

===== Library Menu =====

```

```

1. Add Book Data
2. Search for a Book
3. Display All Books
4. Exit the Program
>> 2

```

```

Enter the title or author of the book you want to search: CALCULUS19

```

```

>>>> Book Record Found!!!

```

```

--- Book Details ---
Book Title: CALCULUS19
Book Author: Alama
Book Genre: educational
Publication Date: 2021-12

```

```

===== Library Menu =====

```

```

1. Add Book Data
2. Search for a Book
3. Display All Books
4. Exit the Program
>> 2

```

```

Enter the title or author of the book you want to search: TOM
No book record found with the search: TOM

```

```

===== Library Menu =====

```

```

1. Add Book Data
2. Search for a Book
3. Display All Books
4. Exit the Program
>> 4

```

```

Exiting the program. Goodbye!

```

```

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home tasks\
24k-2000>

```

Codes Lab-11\_24k-2000 &gt; Problem\_04.c &gt; view\_records()

```

60 void view_records() {
61
62     while (fread(&b[b_count], sizeof(struct library), 1, file)) {
63         b_count++;
64     }
65     fclose(file);
66
67     for (int i = 0; i < b_count; i++) {
68         if (strcmp(b[i].title, search) == 0 || strcmp(b[i].author, search) == 0) {
69             flag = 1;
70             printf("\n\n>>> Book Record Found!!! \n");
71             printf("--- Book Details ---\n");
72             printf("Book Title: %s\n", b[i].title);
73             printf("Book Author: %s\n", b[i].author);
74             printf("Book Genre: %s\n", b[i].genre);
75             printf("Publication Date: %s\n", b[i].pub_date);
76             break;
77         }
78     }
79
80     if (!flag) {
81         printf("No book record found with the search: %s\n", search);
82     }
83 }
84
85 void display_books() {
86     FILE *file = fopen(booksdB, "r");
87     if (file == NULL) {
88         printf("The file does not exist...\n");
89         return;
90     }
91
92     b_count = 0;
93     while (fread(&b[b_count], sizeof(struct library), 1, file)) {
94         b_count++;
95     }
96     fclose(file);
97
98     sort_books();
99
100     printf("==== LIST OF BOOKS ====\n");
101     for (int i = 0; i < b_count; i++) {
102         printf("\n--- Book %d ---\n", i + 1);
103         printf("Title: %s\n", b[i].title);
104         printf("Author: %s\n", b[i].author);
105         printf("Genre: %s\n", b[i].genre);
106         printf("Publication Date: %s\n", b[i].pub_date);
107     }
108 }

```

Codes Lab-11\_24k-2000 &gt; Problem\_04.c &gt; ...

```

120
121 int main() {
122     int choice = 0;
123
124     while (choice != 4) {
125         printf("\n===== Library Menu =====\n");
126         printf("1. Add Book Data\n");
127         printf("2. Search for a Book\n");
128         printf("3. Display All Books\n");
129         printf("4. Exit the Program\n");
130         printf(">> ");
131         scanf("%d", &choice);
132
133         switch (choice) {
134             case 1:
135                 add_data();
136                 break;
137
138             case 2:
139                 view_records();
140                 break;
141
142             case 3:
143                 display_books();
144                 break;
145
146             case 4:
147                 printf("Exiting the program. Goodbye!\n");
148                 break;
149
150             default:
151                 printf("Invalid Input. Please try again.\n");
152                 break;
153         }
154     }
155
156     return 0;
157 }
158

```

## Problem 05;

```

Codes Lab-11_24k-2000 > C Problem_05.c > deposit_withdraw()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  // Create a banking system using structures to store account h
5  // history. Implement the following features:
6  // • Add new accounts.
7  // • Perform deposits and withdrawals.
8  // • Display the account with the highest balance.
9  // • Save and retrieve the account data from a binary file.
10 const char *accountsdb = "accounts.bin";
11
12 struct account {
13     char account_holder[50];
14     char account_number[15];
15     float balance;
16     char transaction_history[5][50];
17 };
18
19 struct account acc[100];
20 int count = 0;
21
22 void add_account();
23 void deposit_withdraw();
24 void display_highest_balance();
25 void save_accounts();
26 void load_accounts();
27
28
29 int main() {
30     int choice = 0;
31
32     load_accounts();
33
34     while (choice != 5) {
35         printf("\n===== Banking System Menu =====\n");
36         printf("1. Add New Account\n");
37         printf("2. Perform Deposit/Withdrawal\n");
38         printf("3. Display Account with Highest Balance\n");
39         printf("4. Save Account Data\n");
40         printf("5. Exit the Program\n");
41         printf(">> ");
42         scanf("%d", &choice);
43

```

```

===== Banking System Menu =====
1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program
>> 1

```

```

===== ADD NEW ACCOUNT =====
Account Holder Name: Muzammil
Account Number: 401231201
Initial Balance: 5000
Account created successfully...

```

```

===== Banking System Menu =====
1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program
>> 1

```

```

===== ADD NEW ACCOUNT =====
Account Holder Name: Fatima
Account Number: 25000
Initial Balance: 100000
Account created successfully...

```

```

===== Banking System Menu =====
1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program
>> 1

```

```

===== ADD NEW ACCOUNT =====
Account Holder Name: Zafar
Account Number: 210021123
Initial Balance: 15000
Account created successfully...

```

```

===== Banking System Menu =====
1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program

```

Codes Lab-11\_24k-2000 &gt; C Problem\_05.c &gt; deposit\_withdraw()

```

29  int main() {
34      while (choice != 5) {
43
44          switch (choice) {
45              case 1:
46                  add_account();
47                  break;
48
49              case 2:
50                  deposit_withdraw();
51                  break;
52
53              case 3:
54                  display_highest_balance();
55                  break;
56
57              case 4:
58                  save_accounts();
59                  break;
60
61              case 5:
62                  printf("Exiting the program. Goodbye!\n");
63                  break;
64
65              default:
66                  printf("Invalid Input. Please try again.\n");
67                  break;
68          }
69      }
70
71      return 0;
72  }
73
74
75
76  void add_account() {
77      printf("\n==== ADD NEW ACCOUNT ==== \n");
78      printf("Account Holder Name: ");
79      scanf(" %s", acc[count].account_holder);
80      printf("Account Number: ");
81      scanf("%s", acc[count].account_number);
82      printf("Initial Balance: ");
83      scanf("%f", &acc[count].balance);
84
85      for (int i = 0; i < 5; i++) {
86          strcpy(acc[count].transaction_history[i], "NULL... No Transaction made yet");
87      }
88
89      printf("Account created successfully...\n");

```

===== Banking System Menu =====

1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program

&gt;&gt; 2

===== DEPOSIT / WITHDRAW =====

Enter Account Number: 210021123

Account Found: Zafar having Balance : 15000.000000

Enter 1 to Deposit and 2 to Withdraw: 1

Enter Deposit Amount: 5000

Amount Deposited Successfully. New Balance: 20000.00

===== Banking System Menu =====

1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program

&gt;&gt; 2

===== DEPOSIT / WITHDRAW =====

Enter Account Number: 25000

Account Found: Fatima having Balance : 100000.000000

Enter 1 to Deposit and 2 to Withdraw: 2

Enter Withdrawal Amount: 50000

Amount Withdrawn Successfully. New Balance: 50000.00

===== Banking System Menu =====

1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program

&gt;&gt; 3

===== ACCOUNT WITH HIGHEST BALANCE =====

Account Holder: Fatima

Account Number: 25000

Balance: 50000.00

===== Banking System Menu =====

1. Add New Account
2. Perform Deposit/Withdrawal
3. Display Account with Highest Balance
4. Save Account Data
5. Exit the Program

&gt;&gt; 4

Account data saved successfully to file.

===== Banking System Menu =====

1. Add New Account
2. Perform Deposit/Withdrawal





Codes Lab-11\_24k-2000 &gt; C Problem\_05.c &gt; ...

```

76 void add_account() {
90     count++;
91 }
92
93 void deposit_withdraw() {
94     char acc_number[15];
95     float amount;
96     int found = 0;
97
98     printf("\n=== DEPOSIT / WITHDRAW ===\n");
99     printf("Enter Account Number: ");
100    scanf("%s", acc_number);
101
102    for (int i = 0; i < count; i++) {
103        if (strcmp(acc[i].account_number, acc_number) == 0) {
104            found = 1;
105            printf("Account Found: %s having Balance : %f\n", acc[i].account_holder, acc[i].balance);
106            printf("Enter 1 to Deposit and 2 to Withdraw: ");
107            int choice;
108            scanf("%d", &choice);
109
110            if (choice == 1) {
111                printf("Enter Deposit Amount: ");
112                scanf("%f", &amount);
113                acc[i].balance += amount;
114
115                snprintf(acc[i].transaction_history[4], 50, "Deposited: %.2f", amount);
116                for (int j = 0; j < 4; j++) {
117                    strcpy(acc[i].transaction_history[j], acc[i].transaction_history[j + 1]);
118                }
119
120                printf("Amount Deposited Successfully. New Balance: %.2f\n", acc[i].balance);
121            } else if (choice == 2) {
122                printf("Enter Withdrawal Amount: ");
123                scanf("%f", &amount);
124                if (amount > acc[i].balance) {
125                    printf("Insufficient Balance. Transaction Failed.\n");
126                } else {
127                    acc[i].balance -= amount;
128
129                    // Update transaction history
130                    snprintf(acc[i].transaction_history[4], 50, "Withdrew: %.2f", amount);
131                    for (int j = 0; j < 4; j++) {
132                        strcpy(acc[i].transaction_history[j], acc[i].transaction_history[j + 1]);
133                    }
134
135                    printf("Amount Withdrawn Successfully. New Balance: %.2f\n", acc[i].balance);
136                }
137            } else {

```

Codes Lab-11\_24k-2000 &gt; C Problem\_05.c &gt; ...

```

93 void deposit_withdraw() {
102     for (int i = 0; i < count; i++) {
103         if (strcmp(acc[i].account_number, acc_number) == 0) {
137             } else {
138                 printf("Invalid Option. Try Again.\n");
139             }
140             break;
141         }
142     }
143
144     if (found == 0) {
145         printf("Account Not Found.\n");
146     }
147 }
148
149 void display_highest_balance() {
150     if (count == 0) {
151         printf("\nNo accounts available.\n");
152         return;
153     }
154
155     int max = 0;
156     for (int i = 1; i < count; i++) {
157         if (acc[i].balance > acc[max].balance) {
158             max = i;
159         }
160     }
161
162     printf("\n=== ACCOUNT WITH HIGHEST BALANCE ===\n");
163     printf("Account Holder: %s\n", acc[max].account_holder);
164     printf("Account Number: %s\n", acc[max].account_number);
165     printf("Balance: %.2f\n", acc[max].balance);
166 }
167
168 void save_accounts() {
169     FILE *file = fopen(accountsdb, "wb");
170     if (file == NULL) {
171         printf("Unable to open file for saving.\n");
172         return;
173     }
174
175     fwrite(&acc, sizeof(struct account), count, file);
176     fclose(file);
177     printf("Account data saved successfully to file.\n");
178 }
179
180 void load_accounts() {
181     FILE *file = fopen(accountsdb, "rb");
182     if (file == NULL) {

```

## Problem 06:

Codes Lab-11\_24k-2000 &gt; C Problem\_06.c &gt; main()

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  // Design a nested structure for employees that stores personal details and a timesheet
5  // for each day of the week (e.g., hours worked). Write a program to Input timesheet data for 10
6  // employees. Calculate and display the total hours worked by each employee in the week. Identify
7  // employees who worked overtime (above 40 hours).
8
9  #define working_days 5
10 #define max 10
11
12 struct daily {
13     float hours_worked_daily;
14 };
15
16 struct timesheet {
17     struct daily d[working_days];
18 };
19
20 struct employees {
21     char employee_id[10];
22     char full_name[20];
23     char gender[10];
24     char dob[15];
25     struct timesheet t_s;
26 };
27
28 struct employees e[max];
29
30 const char *days[working_days] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};
31 int i,j=0;
32 float total_hours_worked[max]={0};
33
34 void input_ts_data(){
35     printf("====INPUT TIMESHEET DATA====\n");
36
37     for (i=0; i<10; i++)
38     {
39         for (j=0; j<5; j++)
40         {
41             printf("Enter hours worked on %s for the Employee %d: ", days[j], i+1);
42             scanf("%f", &e[i].t_s.d[j].hours_worked_daily);
43         }
44         printf("\nSuccessfully Added Timesheet data for Employee %d!!!\n", i+1);
45     }
46 }
47
48 }
49

```

===EMPLOYEE MENU===

1. Input timesheet data for all Employees
2. Calculate and Display Hours worked by an Employee
3. Search for Employees working overtime
4. Exit the Program

&gt;&gt;1

====INPUT TIMESHEET DATA====

Enter hours worked on Monday for the Employee 1: 7  
 Enter hours worked on Tuesday for the Employee 1: 6  
 Enter hours worked on Wednesday for the Employee 1: 5  
 Enter hours worked on Thursday for the Employee 1: 4  
 Enter hours worked on Friday for the Employee 1: 4

Successfully Added Timesheet data for Employee 1!!!  
 Enter hours worked on Monday for the Employee 2: 4  
 Enter hours worked on Tuesday for the Employee 2: 5  
 Enter hours worked on Wednesday for the Employee 2: 6  
 Enter hours worked on Thursday for the Employee 2: 7  
 Enter hours worked on Friday for the Employee 2: 4

Successfully Added Timesheet data for Employee 2!!!  
 Enter hours worked on Monday for the Employee 3: 5  
 Enter hours worked on Tuesday for the Employee 3: 5  
 Enter hours worked on Wednesday for the Employee 3: 3  
 Enter hours worked on Thursday for the Employee 3: 5  
 Enter hours worked on Friday for the Employee 3: 5

Successfully Added Timesheet data for Employee 3!!!  
 Enter hours worked on Monday for the Employee 4: 5  
 Enter hours worked on Tuesday for the Employee 4: 3  
 Enter hours worked on Wednesday for the Employee 4: 2  
 Enter hours worked on Thursday for the Employee 4: 7  
 Enter hours worked on Friday for the Employee 4: 8

Successfully Added Timesheet data for Employee 4!!!  
 Enter hours worked on Monday for the Employee 5: 5  
 Enter hours worked on Tuesday for the Employee 5: 4  
 Enter hours worked on Wednesday for the Employee 5: 3  
 Enter hours worked on Thursday for the Employee 5: 6  
 Enter hours worked on Friday for the Employee 5: 7

Successfully Added Timesheet data for Employee 5!!!  
 Enter hours worked on Monday for the Employee 6: 8  
 Enter hours worked on Tuesday for the Employee 6: 6  
 Enter hours worked on Wednesday for the Employee 6: 5  
 Enter hours worked on Thursday for the Employee 6: 3  
 Enter hours worked on Friday for the Employee 6: 4

Successfully Added Timesheet data for Employee 6!!!  
 Enter hours worked on Monday for the Employee 7: 5  
 Enter hours worked on Tuesday for the Employee 7: 3  
 Enter hours worked on Wednesday for the Employee 7: 5  
 Enter hours worked on Thursday for the Employee 7: 3  
 Enter hours worked on Friday for the Employee 7: 5



Codes Lab-11\_24k-2000 &gt; C Problem\_06.c &gt; main()

```

50 void calculate_display(){
51
52     for (i=0; i<10; i++)
53     {
54         for (j=0; j<5; j++)
55         {
56             total_hours_worked[i] = total_hours_worked[i] + e[i].t_s.d[j].hours_worked_daily;
57         }
58     }
59
60
61     for (i=0; i<10; i++)
62     {
63         printf("Total hours worked for Employee %d throughout the week is : %f\n", i+1, total_hours_worked[i]);
64     }
65 }
66
67 void search_overtime(){
68     printf("====EXEMPLARY EMPLOYEES====\n");
69     int found=0;
70
71     for (i=0; i<10; i++)
72     {
73         if (total_hours_worked[i] > 40)
74         {
75             found = 1;
76             printf("Employee %d has worked overtime upto %.2f hours throughout the entire week !!!\n", i+1, total_hours_worked[i]);
77         }
78     }
79
80     if (found == 0)
81     {
82         printf("No Employees found working overtime...");
83     }
84 }
85
86 int main()
87 {
88     int choice;
89
90     while (choice != 4)
91     {
92         printf("===EMPLOYEE MENU===\n");
93         printf("1. Input timesheet data for all Employees \n");
94         printf("2. Calculate and Display Hours worked by an Employee\n");
95         printf("3. Search for Employees working overtime\n");
96         printf("4. Exit the Program\n");
97         printf(">>>");

```

Successfully Added Timesheet data for Employee 7!!!  
Enter hours worked on Monday for the Employee 8: 3  
Enter hours worked on Tuesday for the Employee 8: 56  
Enter hours worked on Wednesday for the Employee 8: 2  
Enter hours worked on Thursday for the Employee 8: 0  
Enter hours worked on Friday for the Employee 8: 0

Successfully Added Timesheet data for Employee 8!!!  
Enter hours worked on Monday for the Employee 9: 7  
Enter hours worked on Tuesday for the Employee 9: 5  
Enter hours worked on Wednesday for the Employee 9: 3  
Enter hours worked on Thursday for the Employee 9: 5  
Enter hours worked on Friday for the Employee 9: 3

Successfully Added Timesheet data for Employee 9!!!  
Enter hours worked on Monday for the Employee 10: 2  
Enter hours worked on Tuesday for the Employee 10: 2  
Enter hours worked on Wednesday for the Employee 10: 4  
Enter hours worked on Thursday for the Employee 10: 5  
Enter hours worked on Friday for the Employee 10: 6

Successfully Added Timesheet data for Employee 10!!!  
Exiting the Program... Goodbye...==EMPLOYEE MENU==  
1. Input timesheet data for all Employees  
2. Calculate and Display Hours worked by an Employee  
3. Search for Employees working overtime  
4. Exit the Program  
>>2

Total hours worked for Employee 1 throughout the week is : 26.000000  
Total hours worked for Employee 2 throughout the week is : 26.000000  
Total hours worked for Employee 3 throughout the week is : 23.000000  
Total hours worked for Employee 4 throughout the week is : 25.000000  
Total hours worked for Employee 5 throughout the week is : 25.000000  
Total hours worked for Employee 6 throughout the week is : 26.000000  
Total hours worked for Employee 7 throughout the week is : 21.000000  
Total hours worked for Employee 8 throughout the week is : 61.000000  
Total hours worked for Employee 9 throughout the week is : 23.000000  
Total hours worked for Employee 10 throughout the week is : 19.000000  
Exiting the Program... Goodbye...==EMPLOYEE MENU==

1. Input timesheet data for all Employees  
2. Calculate and Display Hours worked by an Employee  
3. Search for Employees working overtime  
4. Exit the Program  
>>3

====EXEMPLARY EMPLOYEES====  
Employee 8 has worked overtime upto 61.00 hours throughout the entire week !!!  
Exiting the Program... Goodbye...==EMPLOYEE MENU==  
1. Input timesheet data for all Employees  
2. Calculate and Display Hours worked by an Employee  
3. Search for Employees working overtime  
4. Exit the Program  
>>4

Invalid Input.... Exiting...Exiting the Program... Goodbye..

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home tasks\Lab-11\_24

```
Codes Lab-11_24k-2000 > C Problem_06.c > main()
86  int main()
87  {
88      int choice;
89
90      while (choice != 4)
91      {
92          printf("===EMPLOYEE MENU===\n");
93          printf("1. Input timesheet data for all Employees \n");
94          printf("2. Calculate and Display Hours worked by an Employee\n");
95          printf("3. Search for Employees working overtime\n");
96          printf("4. Exit the Program\n");
97          printf(">>");
98          scanf("%d", &choice);
99
100         switch (choice)
101         {
102             case 1:
103                 input_ts_data();
104                 break;
105
106             case 2:
107                 calculate_display();
108                 break;
109
110             case 3:
111                 search_overtime();
112                 break;
113
114             default:
115                 printf("Invalid Input.... Exiting...");
116                 break;
117         }
118
119         printf("Exiting the Program... Goodbye..");
120
121     }
122     return 0;
123 }
```

## Problem 07:

Codes Lab-11\_24k-2000 &gt; C Problem\_07.c &gt; ...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  // Create a system to manage hospital patients using nested structures
5  // details, including admission and discharge dates. Calculate the
6  // the hospital. Save patient data into a text file and retrieve it
7
8  const char* patientsdb = "patients.txt";
9
10 struct admission {
11     char admission_date[15];
12 };
13 struct discharge {
14     char discharge_date[15];
15 };
16
17 struct patient {
18     char name[50];
19     char gender[10];
20     int age;
21     struct discharge d_d;
22     struct admission a_d;
23 };
24
25 int c = 0;
26 struct patient *p = NULL;
27
28 void store_data() {
29     c++;
30     p = realloc(p, c * sizeof(struct patient));
31     if (p == NULL) {
32         printf("Memory allocation has failed....\n");
33         return;
34     }
35
36     printf("Enter patient's name: ");
37     scanf(" %s", p[c - 1].name);
38     printf("Enter patient's gender: ");
39     scanf(" %s", p[c - 1].gender);
40     printf("Enter patient's age: ");
41     scanf("%d", &p[c - 1].age);
42     printf("Enter admission date: ");
43     scanf(" %s", p[c - 1].a_d.admission_date);
44     printf("Enter discharge date : ");
45     scanf(" %s", p[c - 1].d_d.discharge_date);
46
47     printf("Patient data added....\n");
48 }

```

```

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home task
Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home task
4k-2000\" ; if ($?) { gcc Problem_07.c -o Problem_07 } ; if (

```

```
==== Patients Management Menu ====
```

1. Store new patient data
2. Calculate Patient's Stay in the Hospital
3. Save data into database
4. Load data from database

```
Enter your choice (-1 to exit): 1
```

```
Enter patient's name: Ali
```

```
Enter patient's gender: male
```

```
Enter patient's age: 12
```

```
Enter admission date: 12/12/2012
```

```
Enter discharge date : 13/12/2012
```

```
Patient data added....
```

```
==== Patients Management Menu ====
```

1. Store new patient data
2. Calculate Patient's Stay in the Hospital
3. Save data into database
4. Load data from database

```
Enter your choice (-1 to exit): 3
```

```
Patient data saved to database
```

```
==== Patients Management Menu ====
```

1. Store new patient data
2. Calculate Patient's Stay in the Hospital
3. Save data into database
4. Load data from database

```
Enter your choice (-1 to exit): 4
```

```
Patient data loaded successfully...
```

```
==== Patients Management Menu ====
```

1. Store new patient data
2. Calculate Patient's Stay in the Hospital
3. Save data into database
4. Load data from database

```
Enter your choice (-1 to exit): 2
```

```
Which patient's stay do you want to view?(Integers only) : 1
```

```
Patient Name : Ali... Days Stayed : 1703832105
```

```
==== Patients Management Menu ====
```

1. Store new patient data
2. Calculate Patient's Stay in the Hospital
3. Save data into database
4. Load data from database

```
Enter your choice (-1 to exit): -1
```

```
Exiting program...
```

```
PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home
b-11_24k-2000>

```



Codes Lab-11\_24k-2000 &gt; C Problem\_07.c &gt; ...

```

50 void calculate_stay() {
51     if (c == 0) {
52         printf("There are noo patients in the database at the moment...");
53         return;
54     }
55
56     printf("Which patient's stay do you want to view?(Integers only) : ");
57     int search;
58     scanf("%d", &search);
59
60     if (search < 1 || search > c) {
61         printf("Invalid patient number.\n");
62         return;
63     }
64
65     int ad_year, ad_month, ad_day;
66     int dd_year, dd_month, dd_day;
67
68     sscanf(p[search-1].a_d.admission_date, "%d-%d-%d", &ad_day, &ad_month, &ad_year);
69     sscanf(p[search-1].d_d.discharge_date, "%d-%d-%d", &dd_day, &dd_month, &dd_year);
70
71     int stay_days = (dd_year - ad_year) * 365 + (dd_month - ad_month) * 30 + (dd_day - ad_day);
72
73     printf("Patient Name : %s... Days Stayed : %d", p[search - 1].name, stay_days);
74 }
75
76 void save_data() {
77     FILE *file = fopen(patientsdb, "w");
78     if (file == NULL) {
79         printf("Error opening file for writing.\n");
80         return;
81     }
82
83     for (int i = 0; i < c; i++) {
84         fprintf(file, "%s,%s,%d,%s,%s\n", p[i].name, p[i].gender, p[i].age, p[i].a_d.admission_date, p[i].d_d.discharge_date);
85     }
86
87     fclose(file);
88     printf("Patient data saved to database\n");
89 }
90
91 void load_data() {
92     FILE *file = fopen(patientsdb, "r");
93     if (file == NULL) {
94         printf("Error.... Can't Open File...\n");
95         return;
96     }
97

```

Codes Lab-11\_24k-2000 &gt; C Problem\_07.c &gt; ...

```
91 void load_data() {
92
93     char buffer[200];
94     c = 0;
95     free(p);
96     p = NULL;
97
98     while (fgets(buffer, sizeof(buffer), file)) {
99         c++;
100         p = realloc(p, c * sizeof(struct patient));
101         if (p == NULL) {
102             printf("Can't Allocate Memory...\n");
103             fclose(file);
104             return;
105         }
106
107         sscanf(buffer, "%[^,],%[^,],%d,%[^,],%s", p[c - 1].name, p[c - 1].gender, &p[c - 1].age, p[c - 1].a_d.admission_date, p[c - 1].d_d.discharge_date);
108     }
109
110     fclose(file);
111     printf("Patient data loaded successfully...\n");
112 }
113
114 int main() {
115     int choice = 0;
116     while (choice != -1) {
117         printf("\n==== Patients Management Menu ==== \n");
118         printf("1. Store new patient data\n");
119         printf("2. Calculate Patient's Stay in the Hospital\n");
120         printf("3. Save data into database\n");
121         printf("4. Load data from database\n");
122         printf("Enter your choice (-1 to exit): ");
123         scanf("%d", &choice);
124
125         switch (choice) {
126             case 1:
127                 store_data();
128                 break;
129
130             case 2:
131                 calculate_stay();
132                 break;
133
134             case 3:
135                 save_data();
136                 break;
137
138             case 4:
139                 load_data();
140                 break;
```

Codes Lab-11\_24k-2000 &gt; C Problem\_07.c &gt; ...

```
119 int main() {
120     int choice = 0;
121     while (choice != -1) {
122         printf("\n==== Patients Management Menu ==== \n");
123         printf("1. Store new patient data\n");
124         printf("2. Calculate Patient's Stay in the Hospital\n");
125         printf("3. Save data into database\n");
126         printf("4. Load data from database\n");
127         printf("Enter your choice (-1 to exit): ");
128         scanf("%d", &choice);
129
130         switch (choice) {
131             case 1:
132                 store_data();
133                 break;
134
135             case 2:
136                 calculate_stay();
137                 break;
138
139             case 3:
140                 save_data();
141                 break;
142
143             case 4:
144                 load_data();
145                 break;
146
147             case -1:
148                 printf("Exiting program...\n");
149                 break;
150
151             default:
152                 printf("Invalid Input... Try again...\n");
153                 break;
154         }
155     }
156
157     free(p);
158
159     return 0;
160 }
161
```

## Problem 08:

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  // Write a program to manage an inventory of products: Use structures to store product
5  // details (ID, name, price, stock quantity). Implement the following:
6  // Add new products.
7  // Search for a product by name or ID.
8  // Update stock quantity after a sale.
9  // Save and load inventory data from a binary file.
10
11 struct products {
12     char product_id[15];
13     char product_name[40];
14     int stock_quantity;
15 };
16
17 const char *productsdb = "products.txt";
18
19 int count = 0;
20 struct products *p = NULL;
21
22 void add_products() {
23     count++;
24     p = realloc(p, count * sizeof(struct products));
25
26     if (p == NULL) {
27         printf("There was an issue with memory allocation...\n");
28         return;
29     }
30
31     printf("==== ADD NEW PRODUCT ==== \n");
32     printf("Enter Product ID: ");
33     scanf("%s", p[count - 1].product_id);
34     printf("Enter Product Name: ");
35     scanf("%s", p[count - 1].product_name);
36     printf("Enter Stock Quantity: ");
37     scanf("%d", &p[count - 1].stock_quantity);
38
39     printf("Product Data Added Successfully...\n");
40 }
41
42 void save_data() {
43     FILE *file = fopen(productsdb, "wb");
44
45     if (file == NULL) {
46         printf("Error opening the file...\n");
47         return;
48     }
49

```

```

p\PF labs Sir Nouman\Labs home tasks\Lab
} ; if ($?) { .\Problem_08 }

```

```
==== Product Management Menu ====
```

1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File

Enter your choice (-1 to exit): 1

```
==== ADD NEW PRODUCT ====
```

Enter Product ID: 12

Enter Product Name: Chips

Enter Stock Quantity: 20

Product Data Added Successfully...

```
==== Product Management Menu ====
```

1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File

Enter your choice (-1 to exit): 1

```
==== ADD NEW PRODUCT ====
```

Enter Product ID: 13

Enter Product Name: oil

Enter Stock Quantity: 25

Product Data Added Successfully...

```
==== Product Management Menu ====
```

1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File

Enter your choice (-1 to exit): 1

```
==== ADD NEW PRODUCT ====
```

Enter Product ID: 05

Enter Product Name: ghee

Enter Stock Quantity: 50

Product Data Added Successfully...

```
==== Product Management Menu ====
```

1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File

Enter your choice (-1 to exit): 2

Enter Product Name or ID to Search: 005

Product not found.

```
==== Product Management Menu ====
```

1. Add New Product

Codes Lab-11\_24k-2000 &gt; C Problem\_08.c &gt; ...

```

42 void save_data() {
43
50     fwrite(p, sizeof(struct products), count, file);
51     fclose(file);
52
53     printf("Product Data Saved Successfully...\n");
54 }
55
56 void load_data() {
57     FILE *file = fopen(productsdb, "rb");
58
59     if (file == NULL) {
60         printf("Error opening the file...\n");
61         return;
62     }
63
64     fseek(file, 0, SEEK_END);
65     int file_size = ftell(file);
66     rewind(file);
67
68     count = file_size / sizeof(struct products);
69     p = realloc(p, count * sizeof(struct products));
70
71     if (p == NULL) {
72         printf("Memory allocation failed while loading data...\n");
73         fclose(file);
74         return;
75     }
76
77     fread(p, sizeof(struct products), count, file);
78     fclose(file);
79
80     printf("Product Data Loaded Successfully...\n");
81 }
82
83 void search_products() {
84     if (count == 0) {
85         printf("No products available to search.\n");
86         return;
87     }
88
89     char search_key[40];
90     printf("Enter Product Name or ID to Search: ");
91     scanf("%s", search_key);
92

```

```

==== Product Management Menu ====
1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File
Enter your choice (-1 to exit): 2
Enter Product Name or ID to Search: 05
Product Found: ID=05, Name=ghee, Stock=50

```

```

==== Product Management Menu ====
1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File
Enter your choice (-1 to exit): 3
==== VIEW PRODUCTS ====
ID: 12, Name: Chips, Stock: 20
ID: 13, Name: oil, Stock: 25
ID: 05, Name: ghee, Stock: 50

```

```

==== Product Management Menu ====
1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File
Enter your choice (-1 to exit): 4
Error opening the file...

```

```

==== Product Management Menu ====
1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File
Enter your choice (-1 to exit): 5
Product Data Saved Successfully...

```

```

==== Product Management Menu ====
1. Add New Product
2. Search for Product by Name or ID
3. View All Products
4. Load Product Data from File
5. Save Product Data to File
Enter your choice (-1 to exit): 4
Product Data Loaded Successfully...

```

```

Codes Lab-11_24k-2000 > C Problem_08.c > ...
83 void search_products() {
92
93     int found = 0;
94     for (int i = 0; i < count; i++) {
95         if (strcmp(p[i].product_id, search_key) == 0 || strcmp(p[i].product_name, search_key) == 0) {
96             printf("Product Found: ID=%s, Name=%s, Stock=%d\n", p[i].product_id, p[i].product_name, p[i].stock_quantity);
97             found = 1;
98         }
99     }
100
101     if (found != 1) {
102         printf("Product not found.\n");
103     }
104 }
105
106 void view_updated() {
107     if (count == 0) {
108         printf("No products available to display.\n");
109         return;
110     }
111
112     printf("==== VIEW PRODUCTS ==== \n");
113     for (int i = 0; i < count; i++) {
114         printf("ID: %s, Name: %s, Stock: %d\n", p[i].product_id, p[i].product_name, p[i].stock_quantity);
115     }
116 }
117
118 int main() {
119     int choice = 0;
120
121     while (choice != -1) {
122         printf("\n==== Product Management Menu ==== \n");
123         printf("1. Add New Product\n");
124         printf("2. Search for Product by Name or ID\n");
125         printf("3. View All Products\n");
126         printf("4. Load Product Data from File\n");
127         printf("5. Save Product Data to File\n");
128         printf("Enter your choice (-1 to exit): ");
129         scanf("%d", &choice);
130
131         switch (choice) {
132             case 1:
133                 add_products();
134                 break;
135
136             case 2:
137                 search_products();
138                 break;
139
140             case 3:
141                 view_updated();
142                 break;
143
144             case 4:
145                 load_data();
146                 break;
147
148             case 5:
149                 save_data();
150                 break;
151
152             case -1:
153                 printf("Exiting program...\n");
154                 break;
155
156             default:
157                 printf("Invalid Input... Try again...\n");
158                 break;
159         }
160     }
161
162     free(p);
163     return 0;
164 }
165

```

```

Codes Lab-11_24k-2000 > C Problem_08.c > ...
117
118 int main() {
119     int choice = 0;
120
121     while (choice != -1) {
122         printf("\n==== Product Management Menu ==== \n");
123         printf("1. Add New Product\n");
124         printf("2. Search for Product by Name or ID\n");
125         printf("3. View All Products\n");
126         printf("4. Load Product Data from File\n");
127         printf("5. Save Product Data to File\n");
128         printf("Enter your choice (-1 to exit): ");
129         scanf("%d", &choice);
130
131         switch (choice) {
132             case 1:
133                 add_products();
134                 break;
135
136             case 2:
137                 search_products();
138                 break;
139
140             case 3:
141                 view_updated();
142                 break;
143
144             case 4:
145                 load_data();
146                 break;
147
148             case 5:
149                 save_data();
150                 break;
151
152             case -1:
153                 printf("Exiting program...\n");
154                 break;
155
156             default:
157                 printf("Invalid Input... Try again...\n");
158                 break;
159         }
160     }
161
162     free(p);
163     return 0;
164 }
165

```



## Problem 09:

Codes Lab-11\_24k-2000 > C Problem\_09.c > findTopPerformer(Player [], int)

```

1  #include <stdio.h>
2  #include <string.h>
3
4  struct Player {
5      char name[50];
6      int runs;
7      int matchesPlayed;
8  };
9  void readDataFromFile(struct Player players[], int *n) {
10     FILE *file = fopen("players.txt", "r");
11     if (file == NULL) {
12         printf("Error opening file.\n");
13         return;
14     }
15
16     while (fscanf(file, "%s %d %d", players[*n].name, &players[*n].runs, &players[*n].matchesPlayed) != EOF) {
17         (*n)++;
18     }
19
20     fclose(file);
21 }
22 void calculateAverageRuns(struct Player players[], int n) {
23     for (int i = 0; i < n; i++) {
24         printf("Player: %s, Average Runs: %.2f\n", players[i].name, (float)players[i].runs / players[i].matchesPlayed);
25     }
26 }
27 void findTopPerformer(struct Player players[], int n) {
28     int bestPlayerIndex = 0;
29     float highestRatio = (float)players[0].runs / players[0].matchesPlayed;
30
31     for (int i = 1; i < n; i++) {
32         float ratio = (float)players[i].runs / players[i].matchesPlayed;
33         if (ratio > highestRatio) {
34             highestRatio = ratio;
35             bestPlayerIndex = i;
36         }
37     }
38     printf("Player with highest runs-to-matches ratio: %s with a ratio of %.2f\n",
39           players[bestPlayerIndex].name, highestRatio);
40 }
41 int main() {
42     struct Player players[50];
43     int n = 0;
44     readDataFromFile(players, &n);
45     calculateAverageRuns(players, n);
46     findTopPerformer(players, n);
47
48     return 0;
49 }

```

```

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home tasks\Lab-11_24k-2000> cd "c:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home tasks\Lab-11_24k-2000\Codes Lab-11_24k-2000\" ; if ($?) { gcc Problem_09.c -o Problem_09 } ; if ($?) { .\Problem_09 }
Error opening file.
Player with highest runs-to-matches ratio: a with a ratio of 1.75
PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nouman\Labs home tasks\Lab-11_24k-2000\Codes Lab-11_24k-2000>

```

## Problem 10:

```

Codes Lab-11_24k-2000 > C Problem_10.c > inputTimetable(Timetable *)
1  #include <stdio.h>
2  #include <string.h>
3  // Create a timetable management system using nested structures:
4  // Store course details, timings, and instructor details.
5  // Input and display the timetable for different departments.
6  // Save the timetable data into a text file.
7  // Allow the user to retrieve and edit the timetable for a specific department
8
9  struct Instructor {
10     char name[50];
11     char department[50];
12 };
13
14 struct Course {
15     char courseName[50];
16     char timing[50];
17     struct Instructor instructor;
18 };
19
20 struct Timetable {
21     char department[50];
22     struct Course courses[10];
23     int numCourses;
24 };
25
26 void inputTimetable(struct Timetable *timetable) {
27     printf("\nEnter details for department: %s\n", timetable->department);
28     printf("Enter number of courses: ");
29     scanf("%d", &timetable->numCourses);
30
31     for (int i = 0; i < timetable->numCourses; i++) {
32         printf("\nCourse %d:\n", i + 1);
33         printf("Course Name: ");
34         scanf(" %s", timetable->courses[i].courseName);
35         printf("Timing: ");
36         scanf(" %s", timetable->courses[i].timing);
37         printf("Instructor Name: ");
38         scanf(" %s", timetable->courses[i].instructor.name);
39         printf("Instructor Department: ");
40         scanf(" %s", timetable->courses[i].instructor.department);
41     }
42 }

```

```

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nou
00> cd "c:\Users\DELL\OneDrive\Desktop\PF labs S
4k-2000\Codes Lab-11_24k-2000\" ; if ($?) { gcc F
($?) { .\Problem_10 }

```

Enter details for department: Computer Science  
Enter number of courses: 3

Course 1:  
Course Name: AppliedPhysics  
Timing: 4:00-6:00  
Instructor Name: SirAdeel  
Instructor Department: AppliedPhysics

Course 2:  
Course Name: Calculus  
Timing: 1:00-3:00  
Instructor Name: SirNadeem  
Instructor Department: Maths

Course 3:  
Course Name: ProgrammingFundamental  
Timing: 10:30-12:30  
Instructor Name: SirNouman  
Instructor Department: EE

Timetable for department: Computer Science

Course: AppliedPhysics  
Timing: 4:00-6:00  
Instructor: SirAdeel, Department: AppliedPhysics

Course: Calculus  
Timing: 1:00-3:00  
Instructor: SirNadeem, Department: Maths

Course: ProgrammingFundamental  
Timing: 10:30-12:30  
Instructor: SirNouman, Department: EE

Enter department name to edit: Calculus  
Department not found.

```

PS C:\Users\DELL\OneDrive\Desktop\PF labs Sir Nou
00\Codes Lab-11_24k-2000>

```

```
Codes Lab-11_24k-2000 > C Problem_10.c > main()
26 void inputTimetable(struct Timetable *timetable) {
42 }
43
44 void displayTimetable(struct Timetable timetable) {
45     printf("\nTimetable for department: %s\n", timetable.department);
46     for (int i = 0; i < timetable.numCourses; i++) {
47         printf("\nCourse: %s\n", timetable.courses[i].courseName);
48         printf("Timing: %s\n", timetable.courses[i].timing);
49         printf("Instructor: %s, Department: %s\n",
50             timetable.courses[i].instructor.name,
51             timetable.courses[i].instructor.department);
52     }
53 }
54 void saveTimetableToFile(struct Timetable timetable) {
55     FILE *file = fopen("timetable.txt", "w");
56     if (file == NULL) {
57         printf("Error opening file.\n");
58         return;
59     }
60     fprintf(file, "Timetable for department: %s\n", timetable.department);
61     for (int i = 0; i < timetable.numCourses; i++) {
62         fprintf(file, "\nCourse: %s\n", timetable.courses[i].courseName);
63         fprintf(file, "Timing: %s\n", timetable.courses[i].timing);
64         fprintf(file, "Instructor: %s, Department: %s\n",
65             timetable.courses[i].instructor.name,
66             timetable.courses[i].instructor.department);
67     }
68     fclose(file);
69 }
70 void editTimetable(struct Timetable *timetable) {
71     printf("\nEnter department name to edit: ");
72     char department[50];
73     scanf("%s", department);
74
75     if (strcmp(timetable->department, department) == 0) {
76         printf("Editing timetable for department: %s\n", timetable->department);
77         inputTimetable(timetable);
78     } else {
79         printf("Department not found.\n");
80     }
81 }
82 int main() {
83     struct Timetable timetable = {"Computer Science", {}, 0};
84     inputTimetable(&timetable);
85     displayTimetable(timetable);
86     saveTimetableToFile(timetable);
87     editTimetable(&timetable);
88     return 0;
89 }
```