

Opening Modes in Standard I/O		
Mode	Meaning of Mode	During Inexistence of file
r	Open for reading.	If the file does not exist, <code>fopen()</code> returns NULL.
rb	Open for reading in binary mode.	If the file does not exist, <code>fopen()</code> returns NULL.
w	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a	Open for append. Data is added to the end of the file.	If the file does not exist, it will be created.
ab	Open for append in binary mode. Data is added to the end of the file.	If the file does not exist, it will be created.
r+	Open for both reading and writing.	If the file does not exist, <code>fopen()</code> returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, <code>fopen()</code> returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exist, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exist, it will be created.

1. C program to read name and marks of n number of students and store them in a file.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      char name[50];
6      int marks, i, num;
7      printf("Enter number of students: ");
8      scanf("%d", &num);
9      FILE *fptr;
10     fptr = (fopen("C:\\student.txt", "w"));
11     if(fptr == NULL)
12     {
13         printf("Error!");
14         exit(1);
15     }
16     for(i = 0; i < num; ++i)
17     {
18         printf("For student%d\nEnter name: ", i+1);
19         scanf("%s", name);
20
21         printf("Enter marks: ");
22         scanf("%d", &marks);
23
24         fprintf(fptr, "\nName: %s \nMarks=%d \n", name, marks);
25     }
26     fclose(fptr);
27     return 0;
28 }

```

input

```

Enter number of students: 2
For student1
Enter name: Mr.X
Enter marks: 72
For student2
Enter name: Mr.A
Enter marks: 85

```

2. C program to write structure to a file using fwrite().

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 struct person
5 {
6     int id;
7     char fname[20];
8     char lname[20];
9 };
10 int main ()
11 {
12     FILE *outfile;
13     // open file for writing
14     outfile = fopen ("person.dat", "w");
15     if (outfile == NULL)
16     {
17         fprintf(stderr, "\nError opened file\n");
18         exit (1);
19     }
20     struct person input1 = {1, "rohit", "sharma"};
21     struct person input2 = {2, "mahendra", "dhoni"};
22
23     // write struct to file
24     fwrite (&input1, sizeof(struct person), 1, outfile);
25     fwrite (&input2, sizeof(struct person), 1, outfile);
26     if(fwrite != 0)
27         printf("contents to file written successfully !\n");
28     else
29         printf("error writing file !\n");
30     // close file
31     fclose (outfile);
32     return 0;
33 }
```

contents to file written successfully !

...Program finished with exit code 0
Press ENTER to exit console.

3. Create a linked list, display its size and update a value in the linked list using C programming language.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node {
5      int data;
6      struct node *next;
7  };
8
9  struct node *head = NULL;
10 struct node *current = NULL;
11
12 //display the list
13 void printList() {
14
15     struct node *ptr = head;
16
17     printf("\n[head] =>");
18     //start from the beginning
19     while(ptr != NULL) {
20         printf(" %d =>", ptr->data);
21         ptr = ptr->next;
22     }
23
24     printf(" [null]\n");
25 }
26
27 //insert link at the first location
28 void insert(int data) {
29     //create a link
30     struct node *link = (struct node*) malloc(sizeof(struct node));
31
32     //link->key = key;
33     link->data = data;
34
35     //point it to old first node

```

```

36     link->next = head;
37
38     //point first to new first node
39     head = link;
40 }
41 void size_of_list() {
42     int size = 0;
43
44     if(head==NULL) {
45         printf("List size : %d ", size);
46         return;
47     }
48
49     current = head;
50     size = 1;
51     while(current->next!=NULL) {
52         current = current->next;
53         size++;
54     }
55     printf("List size : %d ", size);
56 }
57 void update_data(int old, int new) {
58     int pos = 0;
59
60     if(head==NULL) {
61         printf("Linked List not initialized");
62         return;
63     }
64
65     current = head;
66     while(current->next!=NULL) {
67         if(current->data == old) {
68             current->data = new;
69             printf("\n%d found at position %d, replaced with %d\n", old, pos, new);
70             return;

```

```

71     }
72
73     current = current->next;
74     pos++;
75 }
76
77 printf("%d does not exist in the list\n", old);
78 }
79
80
81 int main() {
82     insert(10);
83     insert(20);
84     insert(30);
85     insert(1);
86     insert(40);
87     insert(56);
88
89     printList();
90     size_of_list();
91     update_data(40, 44);
92     printList();
93     return 0;
94 }

```

```

[head] => 56 => 40 => 1 => 30 => 20 => 10 => [null]
List size : 6
40 found at position 1, replaced with 44

[head] => 56 => 44 => 1 => 30 => 20 => 10 => [null]

...Program finished with exit code 0
Press ENTER to exit console.

```

TASKS

1. Write a C program to merge the contents of two files into a third file.
2. Write a C program to read the last line from the file.
3. Write a C program to reverse the contents (character by character) of the file.

<pre> !!uoy knahT .enif ma I ? uoy era woH !!!nuf si gnimmarcorp C ...Program finished with exit code 0 Press ENTER to exit console. </pre>	<table border="1"> <tr> <td>main.c</td> <td>file1.txt</td> <td>:</td> <td>file2.txt</td> </tr> <tr> <td colspan="4"> <pre> 1 C programming is fun!!! 2 How are you ? 3 I am fine. 4 Thank you!! </pre> </td> </tr> </table>	main.c	file1.txt	:	file2.txt	<pre> 1 C programming is fun!!! 2 How are you ? 3 I am fine. 4 Thank you!! </pre>			
main.c	file1.txt	:	file2.txt						
<pre> 1 C programming is fun!!! 2 How are you ? 3 I am fine. 4 Thank you!! </pre>									

4. Write a C program to split the given linked list into even linked list and odd linked list.

Sample Output:

```

Complete list:
[head] => 1 => 2 => 3 => 4 => 5 => 6 => 7 => 8 => 9 => 10 => [null]

Odd : [head] => 1 => 3 => 5 => 7 => 9 => [null]
Even : [head] => 2 => 4 => 6 => 8 => 10 => [null]

```