

## UNIT 4 FILE HANDLING

1. Why files are needed?

Ans: When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.

If you have to enter a large number of data, it will take a lot of time to enter them all.

However, if you have a file containing all the data, you can easily access the contents of the file using few commands in C. You can easily move your data from one computer to another without any changes.

2. What are the different modes of operations performed on a file using fopen() function in C?

Ans: There are many modes for opening a file:

r - open a file in read mode.

w - opens or create a text file in write mode.

a - opens a file in append mode.

r+ - opens a file in both read and write mode.

a+ - opens a file in both read and write mode.

w+ - opens a file in both read and write mode.

3. What are the 3 operations performed on the file?

4. Write a C program to copy the contents from one file to another file

```
/*Program to copy one file to another */
```

```
#include <stdio.h>
```

```
Int main()
```

```
{
```

```
FILE *fp1;
```

```
FILE *fp2;
```

```
int ch;
```

```
if((fp1=fopen("f1.dat","r")) == NULL)
```

```
{
```

```
printf("Error opening input file\n");
```

```
exit(0);
```

```
}
```

```
if((fp2=fopen("f2.dat","w")) == NULL)
```

```
{
```

```
printf("Error opening output file\n");
```

```
exit(0);
```

```
}
```

```
while (!feof(fp1))
```

```
{
```

```
ch=getc(fp1);
```

```
putc(ch,fp2);
```

```
}
```

```
fclose(fp1);
```

```
fclose(fp2);
```

```
}
```

5. What does the following segment of code do

```
fp=fopen("abc.txt", "w");
```

```
fprintf(fp, "hello world!");
```

Ans It writes "hello world!" into the file pointed by fp

6. Write a C program to read and display a text from the file.

Ans. #include <stdio.h>

```
#include <stdlib.h> /* For exit() function*/
```

```
int main()
```

```
{
```

```
char c[1000];
```

```
FILE *fptr;
```

```
if ((fptr=fopen("program.txt","r"))==NULL){
```

```
printf("Error! opening file");
```

```
exit(1); /* Program exits if file pointer returns NULL. */
```

```
}
```

```
fscanf(fptr,"%[^\\n]",c);
```

```
printf("Data from file:\\n%s",c);
```

```
fclose(fptr);
```

```
return 0;
```

```
}
```

7. \_\_\_\_\_ function set the pointer position anywhere in the data file
8. Write a C program to append data in to a file.
9. Suppose we have these statements in a program

```
FILE *fp1, *fp2; char ch;
```

```
fp1 = fopen ("file1", "r"); fp2 = fopen ("file2", "w");
```

Assuming that both files opened successfully, supply the missing arguments in the following function calls:

a) fscanf (\_\_\_\_, "%c", &ch); Ans: fp1

b) fprintf (\_\_\_\_, "%c\\n", ch); Ans: fp2

c) fclose (\_\_\_\_); /\* close the file for writing \*/ Ans: fp2

10. Based on the statement below:

```
fp = fopen ("names.dat", "w");
```

a) What is the statement to declare this file pointer? Ans: FILE \*fp;

b) Write a statement that writes the string "Rock my heart" to the file. Ans: fprintf (fp, "Rock my heart");

c) Write a statement that closes the file. Ans: fclose (fp);

11. What does this program do?

```
int main()
```

```
{
```

```
FILE *fp;
```

```
char str[80];
```

```
fp = fopen ("file.txt", "w");
```

```
if (fp == NULL)
```

```
{
```

```
printf("Error opening file.\\n");
```

```
return 1;
```

```
}
```

```
do
```

```
{
```

```
printf ("Enter a string (ENTER to quit): ");
```

```

gets (str);
(str, "\n");
fputs (str,fp);
strcat
} while (*str != '\n');
return 0;
}

```

Ans. Concatenates the string entered by the user with a <newline>. Writes str to the file until the user presses ENTER.

12. Write a C program to display the contents of the file in reverse order.
13. Write a C program to count no.of characters, spaces, lines, words of a file.
14. Write a C Program to check whether a file can be opened for reading.
15. The following loop appears in fcopy.c program.

```

while((ch=getc(source_fp)) !=EOF)
putc(ch, dest_fp);

```

suppose that we neglected to put parentheses around ch=getc(source\_fp):

```

while(ch=getc(source_fp) !=EOF)
putc(ch, dest_fp);

```

would the program compile without any error? If so, what would the program do when it's run.

16. Find the error in the following function and show how to fix it.

```

int count_periods(const char *filename)
{
FILE *fp;
int n=0;
if((fp = fopen(filename, "r"))!= NULL) {
while (fgetc(fp) != EOF)
if (fgetc(fp) == '.')
n++;
fclose(fp);
}
return(n);
}

```

17. Write short notes on fseek(),ftell(),rewind(),fread(),fprintf(),feof() and ferror()
18. The function used for writing a character to a file is \_\_\_\_\_
19. The mode used for opening an existing file for reading & writing a text stream is \_\_\_\_\_
20. When a program is terminated; all the files used by it are automatically closed. Why is it necessary to close a file during execution of program? Discuss.
21. Explain perror() and strerror().
22. Distinguish between the following functions.
  - a) getc() and getchar().
  - b) scanf() and fscanf().
  - c) printf() and fprintf().
  - d) feof() and ferror().
23. Identify the difference between append and write mode.
24. Write a C program to read name and marks of n number of students from user and store them in a file.

Ans: #include <stdio.h>

```

#include<conio.h>
int main()
{
    char name[50];
    int marks,i,n;
    printf("Enter number of students: ");
    scanf("%d",&n);
    FILE *fptr;
    fptr=(fopen("C:\\student.txt","w"));
    if(fptr==NULL)
    {
        printf("Error!");
    }
    for(i=0;i<n;++i)
    {
        printf("For student%d\nEnter name: ",i+1);
        scanf("%s",name);
        printf("Enter marks: ");
        scanf("%d",&marks);
        fprintf(fptr,"\nName: %s \nMarks=%d \n",name,marks);
    }
    fclose(fptr);
    getch();
    return 0;
}

```

25. Write a C program to write all the members of an array of structures to a file using fwrite().  
Read the array from the file and display on the screen.

```

#include<stdio.h>
#include<conio.h>
struct s
{
    char name[50];
    int height;
};
int main()
{
    struct s a[5],b[5];
    FILE *fptr;
    int i;
    fptr=fopen("file.txt","w");
    for(i=0;i<5;++i)
    {
        fflush(stdin);
        printf("Enter name: ");
        gets(a[i].name);
        printf("Enter height: ");
        scanf("%d",&a[i].height);
    }
}

```

```

fwrite(a,sizeof(a),1,fptr);
fclose(fptr);
fptr=fopen("file.txt","r");
fread(b,sizeof(b),1,fptr);
for(i=0;i<5;++i)
{
printf("Name: %s\nHeight: %d",b[i].name,b[i].height);
}
fclose(fptr);
getch(); }

```

26. What is searching?

Ans: Searching is the process of finding a given value position in a list of values. It decides whether a search key is present in the data or not. It is the algorithmic process of finding a particular item in a collection of items.

Types of searching techniques:

i) Linear Search

ii) Binary Search

27. Explain the working of binary search algorithm.

Ans: 1. It starts with an assumption of basically ignoring the half of the elements just after one comparison.

2. Then perform comparison of x with the middle element.

3. If x matches with middle element, it returns the mid index.

4. Else If x is greater than the mid element, then x can be present in right half of subarray after the mid element. So, search perform on right half.

5. Else (x is smaller) then search perform for the left half.

28. Write pseudo code to search the array in the reverse order, returning 0 when the element is not found.

Ans. 1. Set i to n.

2. Repeat this loop

a. If  $i \leq 0$ , then  $exit = ""$  the  $=""$

b. If  $A[i] = x$ , then exit the loop.

c. Set i to i - 1.

3. Return i

29. Write a C program to implement recursive binary search.

Ans. #include <stdio.h>

// A recursive binary search function. It returns location of x in given //array arr[l..r] is present, // otherwise -1

```
int binarySearch(int arr[], int l, int r, int x)
```

```
{
```

```
if (r >= l) {
```

```
int mid = l + (r - l) / 2;
```

```
// If the element is present at the middle
```

```
// itself
```

```
if (arr[mid] == x)
```

```
return mid;
```

```
// If element is smaller than mid, then
```

```
// it can only be present in left subarray
```

```
if (arr[mid] > x)
```

```

return binarySearch(arr, l, mid - 1, x);
// Else the element can only be present
// in right subarray
return binarySearch(arr, mid + 1, r, x);
}
// We reach here when element is not
// present in array
return -1;
}
int main(void)
{
int arr[] = { 2, 3, 4, 10, 40 };
int n = sizeof(arr) / sizeof(arr[0]);
int x = 10;
int result = binarySearch(arr, 0, n - 1, x);
(result == -1) ? printf("Element is not present in array")
: printf("Element is present at index %d",
result);
return 0;
}

```

30. What is selection sort?

Ans. The Selection sort algorithm is based on the idea of finding the minimum or maximum element in an unsorted array and then putting it in its correct position in a sorted array.

31. Perform Selection sort the following array. Show the array after each swap that takes place.  
{ 30, 60, 20, 50, 40, 10 }.

Ans. 30 60 20 50 40 10

32. Explain the working of iterative binary search method.

Ans. Iterative call is looping over the same block of code multiple times

```

int mysearch(int a[],int low, int high, int key)
{ int pos = -1;
int found = 0;
//if found variable is not created, is there any problem?
while(low<=high && found ==0)
{ int mid = (low+high)/2;
if(a[mid]==key)
{ pos = mid; found = 1; }
else if(key<a[mid])
high = mid-1;
else
low = mid+1;
}
return pos;
}

```

33. What is the difference between Function Pointer and Callback Functions?

34. The given data in matches.csv is sorted as per the year. Write a function to sort the data using city and write it to a new csv file.