1- Write a C program to find properties of a file using stat() function.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/stat.h>
#include <time.h>


void printFileProperties(struct stat stats);


int main()
{
    char path[100];
    struct stat stats;

    printf("Enter source file path: ");
    scanf("%s", path);


    // stat() returns 0 on successful operation,
    // otherwise returns -1 if unable to get file properties.
    if (stat(path, &stats) == 0)
    {
        printFileProperties(stats);
    }
    else
    {
        printf("Unable to get file properties.\n");
        printf("Please check whether '%s' file exists.\n", path);
    }

    return 0;
}
```

```c
/**
 * Function to print file properties.
 */
void printFileProperties(struct stat stats)
{
    struct tm dt;

    // File permissions
    printf("\nFile access: ");
    if (stats.st_mode & R_OK)
        printf("read ");
    if (stats.st_mode & W_OK)
        printf("write ");
    if (stats.st_mode & X_OK)
        printf("execute");

    // File size
    printf("\nFile size: %d", stats.st_size);

    // Get file creation time in seconds and
    // convert seconds to date and time format
    dt = *(gmtime(&stats.st_ctime));
    printf("\nCreated on: %d-%d-%d %d:%d:%d", dt.tm_mday, dt.tm_mon,
dt.tm_year + 1900, dt.tm_hour, dt.tm_min, dt.tm_sec);

    // File modification time
    dt = *(gmtime(&stats.st_mtime));
    printf("\nModified on: %d-%d-%d %d:%d:%d", dt.tm_mday, dt.tm_mon,
dt.tm_year + 1900,dt.tm_hour, dt.tm_min, dt.tm_sec);

}
```

Output:

Output

```
Enter source file path:
data/file3.txt

File access: read write
File size: 115
Created on: 4-1-2018
16:34:13
Modified on: 5-2-2018
19:1:10
```

2. Write a program in C to find the content of the file and number of lines in a Text File.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define LSIZ 128
#define RSIZ 10

int main(void)
{
    char line[RSIZ][LSIZ];
      char fname[20];
    FILE *fptr = NULL;
    int i = 0;
    int tot = 0;
    printf("\n\n Find the content of the file and number of lines in a Text
File :\n");
      printf("-------------------------------------------------------------
-------\n");
      printf(" Input the file name to be opened : ");
      scanf("%s",fname);

    fptr = fopen(fname, "r");
/*---------------------- store the lines into an array ----------------*/
    while(fgets(line[i], LSIZ, fptr))
      {
        line[i][strlen(line[i]) - 1] = '\0';
        i++;
    }
    tot = i;
      printf("\n The content of the file %s  are : \n",fname);
    for(i = 0; i < tot; ++i)
    {
        printf(" %s\n", line[i]);
    }
/*-----------------------------------------------------------------------*/
    printf("\n The lines in the file are : %d\n",tot-1);
    printf("\n");

    return 0;
}
```

3. Write a program in C to delete a specific line from a file.

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

#define MAX 256

  int main()
  {
        int lno, ctr = 0;
        char ch;
        FILE *fptr1, *fptr2;
            char fname[MAX];
        char str[MAX], temp[] = "temp.txt";
            printf("\n\n Delete a specific line from a file :\n");
            printf("-----------------------------------------\n");
            printf(" Input the file name to be opened : ");
            scanf("%s",fname);
        fptr1 = fopen(fname, "r");
        if (!fptr1)
            {
                printf(" File not found or unable to open the input file!!\n");
                return 0;
        }
        fptr2 = fopen(temp, "w"); // open the temporary file in write mode
        if (!fptr2)
            {
                printf("Unable to open a temporary file to write!!\n");
                fclose(fptr1);
                return 0;
        }
        printf(" Input the line you want to remove : ");
        scanf("%d", &lno);
            lno++;
        // copy all contents to the temporary file except the specific line
        while (!feof(fptr1))
        {
            strcpy(str, "\0");
            fgets(str, MAX, fptr1);
            if (!feof(fptr1))
            {
                ctr++;
                /* skip the line at given line number */
                if (ctr != lno)
                {
                    fprintf(fptr2, "%s", str);
                }
            }
        }
```

```
        fclose(fptr1);
        fclose(fptr2);
        remove(fname);              // remove the original file
        rename(temp, fname);  // rename the temporary file to original
name
/*------ Read the file ----------------*/
   fptr1=fopen(fname,"r");
           ch=fgetc(fptr1);
         printf(" Now the content of the file %s is : \n",fname);
         while(ch!=EOF)
           {
               printf("%c",ch);
                ch=fgetc(fptr1);
           }
        fclose(fptr1);
/*------- End of reading ---------------*/
        return 0;

  }
```

4. Write a program in C to replace a specific line with another text in a file.

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

#define MAX 256

  int main()
  {
        FILE *fptr1, *fptr2;
        int lno, linectr = 0;
        char str[MAX],fname[MAX];
        char newln[MAX], temp[] = "temp.txt";

            printf("\n\n Replace a specific line in a text file with a new text
:\n");
            printf("-------------------------------------------------------------
-\n");
            printf(" Input the file name to be opened : ");
        fgets(fname, MAX, stdin);
        fname[strlen(fname) - 1] = '\0';
        fptr1 = fopen(fname, "r");
        if (!fptr1)
        {
                printf("Unable to open the input file!!\n");
                return 0;
        }
        fptr2 = fopen(temp, "w");
        if (!fptr2)
        {
                printf("Unable to open a temporary file to write!!\n");
                fclose(fptr1);
                return 0;
        }
        /* get the new line from the user */
        printf(" Input the content of the new line : ");
        fgets(newln, MAX, stdin);
        /* get the line number to delete the specific line */
        printf(" Input the line no you want to replace : ");
        scanf("%d", &lno);
        lno++;
         // copy all contents to the temporary file other except specific line
```

```
while (!feof(fptr1))
        {
            strcpy(str, "\0");
            fgets(str, MAX, fptr1);
            if (!feof(fptr1))
            {
                linectr++;
                if (linectr != lno)
                    {
                        fprintf(fptr2, "%s", str);
                    }
                    else
                    {
                        fprintf(fptr2, "%s", newln);
                    }
                }
        }
        fclose(fptr1);
        fclose(fptr2);
        remove(fname);
        rename(temp, fname);
        printf(" Replacement did successfully..!! \n");
        return 0;
  }
```