

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

c prog 19-5 home assignment

HA10.1 WAP which produces its own source code as its output.

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

```
int main() {
```

```
    FILE *fp_3260;
```

```
    int c;
```

```
    fp_3260 = fopen(__FILE__, "r");
```

```
    do {
```

```
        c = getc(fp_3260);
```

```
        putchar(c);
```

```
    }
```

```
    while(c != EOF);
```

```
    fclose(fp_3260);
```

```
    return 0;
```

```
}
```

```
1#include <stdio.h>
2int main() {
3    FILE *fp_3260;
4    int c;
5
6
7    fp_3260= fopen(__FILE__, "r");
8
9    do {
10        c = getc(fp_3260);
11        putchar(c);
12    }
13    while(c != EOF);
14
15    fclose(fp_3260);
16    return 0;
17}
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ gedit sourceCode.c
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ gcc sourceCode.c
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ ./a.out
#include <stdio.h>
int main() {
    FILE *fp_3260;
    int c;

    fp_3260= fopen(__FILE__,"r");

    do {
        c = getc(fp_3260);
        putchar(c);
    }
    while(c != EOF);

    fclose(fp_3260);
    return 0;
}
```

HA10.2 WAP to copy the specified bytes from given offset of a file and write them into another file.

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

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

```
#include <stdlib.h>
```

```
int main(int argc,char *argv[])
```

```
{
```

```
    FILE *fp1_3260;
```

```
    FILE *fp2_3260;
```

```
    int count    = 0;
```

```
    int location  = 0;
```

```
    int totBytes  = 0;
```

```
    char filename[100], filename1[100];
```

```
    printf("Enter the filename you want copy: ");
```

```
    scanf("%s", filename);
```

```
    printf("Enter the filename you want it copied it to: ");
```

```
    scanf("%s", filename1);
```

```
    unsigned char data[1024];
```

```
    if( argc < 5 )
```

```
    {
```

```
        printf("Insufficient Arguments!!!\n");
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
    printf("Please use \"program-name source-file-name target-file-name offset  
N\" format.\n");  
    return -1;  
}
```

```
fp1_3260 = fopen(filename,"r");  
if( fp1_3260 == NULL )  
{  
    printf("\n File can not be opened : \n");  
    return -1;  
}
```

```
fseek(fp1_3260,0,SEEK_END);
```

```
count  = ftell(fp1_3260);  
location = atoi(argv[3]);    // offset of source file to copy  
totBytes = atoi(argv[4]);    // number of bytes to copy
```

```
if( count < (location + totBytes) )  
{  
    printf("\nGiven number of bytes can not be copy, due to file size.\n");  
    return -1;  
}
```

```
fp2_3260 = fopen(filename1,"w");  
if( fp2_3260 == NULL )  
{  
    printf("\n File can not be opened\n");  
    return -1;  
}
```

```
fseek(fp1_3260,location,SEEK_SET);
```

```
fread(data,totBytes,1,fp1_3260);
```

```
fwrite(data,totBytes,1,fp2_3260);
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
data[totBytes]=0;
```

```
printf("\nCopied content is : \"%s\"\n",data);
```

```
fclose(fp1_3260);
```

```
fclose(fp2_3260);
```

```
return 0;
```

```
}
```

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 int main(int argc,char *argv[])
6 {
7     FILE *fp1;
8     FILE *fp2;
9
10    int count      = 0;
11    int location    = 0;
12    int totBytes    = 0;
13    char filename[100], filename1[100];
14    printf("Enter the filename you want copy: ");
15    scanf("%s", filename);
16
17    printf("Enter the filename you want it copied it to: ");
18    scanf("%s", filename1);
19    unsigned char data[1024];
20
21    if( argc < 5 )
22    {
23        printf("Insufficient Arguments!!!\n");
24        printf("Please use \"program-name source-file-name target-file-name\n");
25        printf("offset N\" format.\n");
26        return -1;
27    }
28
29    fp1 = fopen(filename,"r");
30    if( fp1 == NULL )
31    {
32        printf("\n File can not be opened : \n");
33        return -1;
34    }
35
36    fseek(fp1,0,SEEK_END);
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
34
35     fseek(fp1,0,SEEK_END);
36
37     count    = ftell(fp1);
38     location = atoi(argv[3]);    // offset of source file to copy
39     totBytes = atoi(argv[4]);    // number of bytes to copy
40
41     if( count < (location + totBytes) )
42     {
43         printf("\nGiven number of bytes can not be copy, due to file size.\n");
44         return -1;
45     }
46
47     fp2 = fopen(filename1,"w");
48     if( fp2 == NULL )
49     {
50         printf("\n File can not be opened\n");
51         return -1;
52     }
53
54     fseek(fp1,location,SEEK_SET);
55
56     fread(data,totBytes,1,fp1);
57
58     fwrite(data,totBytes,1,fp2);
59
60     data[totBytes]=0;
61
62     printf("\nCopied content is : \"%s\"\n",data);
63
64     fclose(fp1);
65     fclose(fp2);
66
67     return 0;
68 }
69
```

```
motolomygolda@motolomygolda:~/Projects/Lab_exercises/files$ ./a.out t1.txt t3.txt 0 30
Enter the filename you want copy: t1.txt
Enter the filename you want it copied it to: t3.txt

Copied content is : "This Is One Of The Best Moment"
motolomygolda@motolomygolda:~/Projects/Lab_exercises/files$
```

HA10.3 Write a menu driven program to perform the following operations in a file.

- to delete a specific line from a text file.
- To replace a specified line by another from a text file.

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

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

```
#define MAX 256
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
int main()
{
    FILE *fptr1_3260, *fptr2_3260;
    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_3260 = fopen(fname, "r");
    if (!fptr1_3260)
    {
        printf("Unable to open the input file!!\n");
        return 0;
    }
    fptr2_3260 = fopen(temp, "w");
    if (!fptr2_3260)
    {
        printf("Unable to open a temporary file to write!!\n");
        fclose(fptr1_3260);
        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_3260))
    {
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
        strcpy(str, "\0");
        fgets(str, MAX, fptr1_3260);
        if (!feof(fptr1_3260))
        {
            linectr++;
            if (linectr != lno)
            {
                fprintf(fptr2_3260, "%s", str);
            }
            else
            {
                fprintf(fptr2_3260, "%s", newln);
            }
        }
    }
    fclose(fptr1_3260);
    fclose(fptr2_3260);
    remove(fname);
    rename(temp, fname);
    printf(" Replacement did successfully...!! \n");
    return 0;
}
```



NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
1#include <stdio.h>
2#include <string.h>
3
4#define MAX 256
5
6int main()
7{
8    FILE *fptr1_3260, *fptr2_3260;
9    int lno, linectr = 0;
10    char str[MAX], fname[MAX];
11    char newln[MAX], temp[] = "temp.txt";
12
13    printf("\n\n Replace a specific line in a text file with a new
text :\n");
14    printf("-----\n");
15    printf(" Input the file name to be opened : ");
16    fgets(fname, MAX, stdin);
17    fname[strlen(fname) - 1] = '\0';
18    fptr1_3260 = fopen(fname, "r");
19    if (!fptr1_3260)
20    {
21        printf("Unable to open the input file!!\n");
22        return 0;
23    }
24    fptr2_3260 = fopen(temp, "w");
25    if (!fptr2_3260)
26    {
27        printf("Unable to open a temporary file to write!!\n");
28        fclose(fptr1_3260);
29        return 0;
30    }
31    /* get the new line from the user */
32    printf(" Input the content of the new line : ");
33    fgets(newln, MAX, stdin);
34    /* get the line number to delete the specific line */
35    printf(" Input the line no you want to replace : ");
36    scanf("%d", &lno);
37
38    fgetc(newln, MAX, stdin);
39    /* get the line number to delete the specific line */
40    printf(" Input the line no you want to replace : ");
41    scanf("%d", &lno);
42    lno++;
43    // copy all contents to the temporary file other except specific line
44    while (!feof(fptr1_3260))
45    {
46        strcpy(str, "\0");
47        fgets(str, MAX, fptr1_3260);
48        if (!feof(fptr1_3260))
49        {
50            linectr++;
51            if (linectr != lno)
52            {
53                fprintf(fptr2_3260, "%s", str);
54            }
55            else
56            {
57                fprintf(fptr2_3260, "%s", newln);
58            }
59        }
60    }
61    fclose(fptr1_3260);
62    fclose(fptr2_3260);
63    remove(fname);
64    rename(temp, fname);
65    printf(" Replacement did successfully...!! \n");
66    return 0;
67 }
```



NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ ./a.out

Replace a specific line in a text file with a new text :
-----
Input the file name to be opened : student.txt
Input the content of the new line : Roll number: 21053260
Input the line no you want to replace : 4
Replacement did successfully..!!
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$
```

```
1
2 Name: golda
3 Marks=98
4 Gender: male
5 Roll number: 21053260
6
7 Name: helen
8 Marks=99
9 gender: female
10
```

HA10.4 WAP to merge lines alternatively from two files & print the result.

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

```
#include <stdlib.h>
```

```
void main()
```

```
{
```

```
    FILE *fold1, *fold2, *fnew;
```

```
    char ch, fname1_3260[20], fname2_3260[20], fname3_3260[30];
```

```
    printf("\n\n Merge two files and write it in a new file :\n");
```

```
    printf("-----\n");
```

```
    printf(" Input the 1st file name : ");
```

```
    scanf("%s",fname1_3260);
```

```
    printf(" Input the 2nd file name : ");
```

```
    scanf("%s",fname2_3260);
```

```
    printf(" Input the new file name where to merge the above two files : ");
```

```
    scanf("%s",fname3_3260);
```

```
    fold1=fopen(fname1_3260, "r");
```

```
    fold2=fopen(fname2_3260, "r");
```

```
    if(fold1==NULL || fold2==NULL)
```

```
{
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
//          perror("Error Message ");
           printf(" File does not exist or error in opening...!!\n");
           exit(EXIT_FAILURE);
       }
       fnew=fopen(fname3_3260, "w");
       if(fnew==NULL)
       {
//          perror("Error Message ");
           printf(" File does not exist or error in opening...!!\n");
           exit(EXIT_FAILURE);
       }
       while((ch=fgetc(fold1))!=EOF)
       {
           fputc(ch, fnew);
       }
       while((ch=fgetc(fold2))!=EOF)
       {
           fputc(ch, fnew);
       }
       printf(" The two files merged into %s file successfully..!!\n\n",
fname3_3260);
       fclose(fold1);
       fclose(fold2);
       fclose(fnew);
   }
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
1#include <stdio.h>
2#include <stdlib.h>
3void main(){
4    FILE *fold1, *fold2, *fnew;
5    char ch, fname1_3260[20], fname2_3260[20], fname3_3260[30];
6    printf("\n\n Merge two files and write it in a new file :\n");
7    printf("-----\n");
8    printf(" Input the 1st file name : ");
9    scanf("%s",fname1_3260);
10   printf(" Input the 2nd file name : ");
11   scanf("%s",fname2_3260);
12   printf(" Input the new file name where to merge the above two files : ");
13   scanf("%s",fname3_3260);
14   fold1=fopen(fname1_3260, "r");
15   fold2=fopen(fname2_3260, "r");
16   if(fold1==NULL || fold2==NULL){
17       printf(" File does not exist or error in opening...!!\n");
18       exit(EXIT_FAILURE);
19   }
20   fnew=fopen(fname3_3260, "w");
21   if(fnew==NULL)
22   {
23       printf(" File does not exist or error in opening...!!\n");
24       exit(EXIT_FAILURE);
25   }
26   while((ch=fgetc(fold1))!=EOF)
27   {
28       fputc(ch, fnew);
29   }
30   while((ch=fgetc(fold2))!=EOF){
31       fputc(ch, fnew);
32   }
33   printf(" The two files merged into %s file successfully...!!\n\n",
34   fname3_3260);
35   fclose(fold1);
36   fclose(fold2);
37   fclose(fnew);
38 }
```

```
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ ./a.out
```

```
 Merge two files and write it in a new file :
```

```
-----
Input the 1st file name : source1.txt
```

```
Input the 2nd file name : source2.txt
```

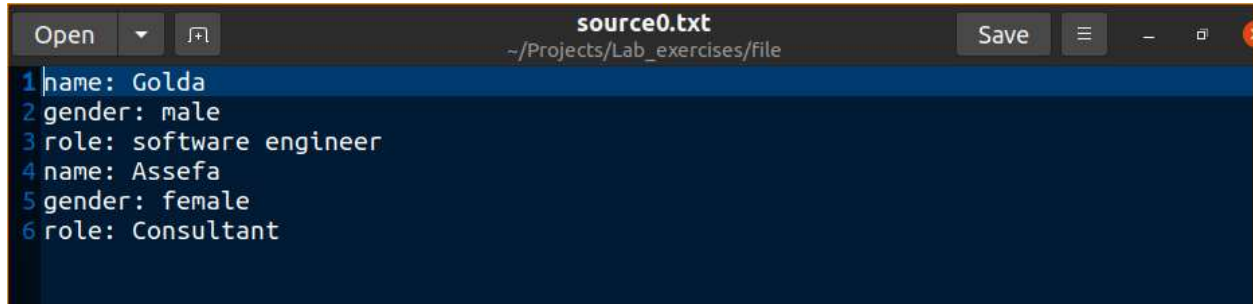
```
Input the new file name where to merge the above two files : source0.txt
```

```
The two files merged into source0.txtttttttttttttt file successfully...!!
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

A screenshot of a code editor window with a dark theme. The title bar shows 'source0.txt' and the file path '~/.Projects/Lab\_exercises/file'. The editor contains a list of six entries, each with a number and a colon. The first three entries are '1 name: Golda', '2 gender: male', and '3 role: software engineer'. The next three entries are '4 name: Assefa', '5 gender: female', and '6 role: Consultant'. The text is in a light blue color on a dark blue background.

```
1 name: Golda
2 gender: male
3 role: software engineer
4 name: Assefa
5 gender: female
6 role: Consultant
```

HA10.5 WAP to display the files and sub-directories present in a specified directory

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

```
#include <dirent.h>
```

```
int main(void)
```

```
{
```

```
    struct dirent *de;
```

```
    DIR *dr_3260 = opendir(".");
```

```
    if (dr_3260 == NULL)
```

```
    {
```

```
        printf("Could not open current directory" );
```

```
        return 0;
```

```
    }
```

```
    while ((de = readdir(dr_3260)) != NULL)
```

```
        printf("%s\n", de->d_name);
```

```
    closedir(dr_3260);
```

```
    return 0;
```

```
}
```

NAME: KIDUS ABEBE

ROLL NUMBER: 2106285

BRANCH: IT

```
1#include <stdio.h>
2#include <dirent.h>
3
4int main(void)
5{
6    struct dirent *de;
7
8    DIR *dr_3260 = opendir(".");
9
10   if (dr_3260 == NULL)
11   {
12       printf("Could not open current directory" );
13       return 0;
14   }
15   while ((de = readdir(dr_3260)) != NULL)
16       printf("%s\n", de->d_name);
17
18   closedir(dr_3260);
19   return 0;
20}
```

```
motolomygolda@motolomygolda:~/Projects/Lab_exercises/file$ ./a.out
source1.txt
studentsFile.c
.
source0.txt
delete.c
evenOdd.txt
filename
source2.txt
sourceCode.c
rev.c
```