Roll no:s201805063

OBSERVATIONS ON R+ MODE IN FILE OPERATION

#1. If one file pointer is used for both reading from the file and writing in it ,then after reading it writes after the existing text. After closing the file the new txt is written after the previous text

Cause:(After reading from the existing file the file pointer reaches at the end and starts writing from there)

Codes: #include<stdio.h> //the existing file is reversed in the program

```
int main()
{
   FILE *fp,*fpw1,*fpw2;
   fp=fopen("my_file.txt","r+");
   //fpw1=fopen("my_file.txt","r+");
   char ch[10][10];
   int i=0,j=0;
   while((fscanf(fp,"%s",ch[i]))!=EOF)
   {
      printf(ch[i]);
      i++;
   }
   int pos=ftell(fp);
   //fseek(fp,0,0);
   while(i>=0)
   {
      i--;
      fprintf(fp,"%s\t",ch[i]);
   }
   fclose(fp);
  // fclose(fpw1);
   return 0;}
```

2.If one file pointer is used for both reading from the file and writing in it.After reading if we use fseek(fp,0,0) then use the pointer to write something. The existing text is erased and the new text is written.

Causes:when we use fseek(fp,0,0) the file pointer moves to the beginning of the file and then starts writing from there almost like a pointer opened for writing(mode ”w”) on the same file.

Codes: #include<stdio.h>

```
int main()
{
   FILE *fp,*fpw1,*fpw2;
   fp=fopen("my_file.txt","r+");
   //fpw1=fopen("my_file.txt","r+");
   char ch[10][10];
   int i=0,j=0;
   while((fscanf(fp,"%s",ch[i]))!=EOF)
   {
      printf(ch[i]);
      i++;
   }
   int pos=ftell(fp);
   fseek(fp,0,0);
   while(i>=0)
   {
      i--;
      fprintf(fp,"%s\t",ch[i]);
   }
   fclose(fp);
  // fclose(fpw1);
 return 0;
}
```

//the output gives one garbage character at the end

3.If two file pointer is used one for reading and another writing. After closing the file the existing file erases and new text is written from the beginning.

Cause: As different pointers are used to read and write after reading the writing pointer is still at the start of the file so when we call it to write it start's writing from the beginning almost like a pointer opened for writing.

Codes:

```
#include<stdio.h>

int main()

{

   FILE *fp,*fpw1,*fpw2;

   fp=fopen("my_file.txt","r+");

   fpw1=fopen("my_file.txt","r+");

   char ch[10][10];

   int i=0,j=0;

   while((fscanf(fp,"%s",ch[i]))!=EOF)

   {

      printf(ch[i]);

      i++;

   }

   int pos=ftell(fp);

  // fseek(fp,0,0);

   while(i>=0)

   {

     i--;

      fprintf(fpw1,"%s\t",ch[i]);

   }

   fclose(fp);

   fclose(fpw1);

 return 0;

}
```

4.If two file pointer is used one for reading and another writing. After reading we find the last position of the file then using the ftell() function and we move the second pointer to that position using fseek().then we call it to write something in the file. After closing the file the existing file erases and new text is written from the beginning.

Cause: As no character is read from the existing file with the second pointer(there is no data before the writing starts) then while writing, the whole file erases and writes the new file

Codes:

```
#include<stdio.h>

int main()

{

   FILE *fp,*fpw1,*fpw2;

   fp=fopen("my_file.txt","r+");

   fpw1=fopen("my_file.txt","r+");

   char ch[10][10];

   int i=0,j=0;

   while((fscanf(fp,"%s",ch[i]))!=EOF)

   {   printf(ch[i]);

      i++;}

   int pos=ftell(fp);

   fseek(fpw1,0,pos);

   while(i>=0)

   { i--;

      fprintf(fpw1,"%s\t",ch[i]); }

   fclose(fp);

   fclose(fpw1);

 return 0;}
```

#5. fread() and fwrite() can not be used as the file is not opened in binary mode.so when we use them it crashes the program.