۱ الف)

a)

```
    if (fputs("Hello World!\n",file) >= 0 ){
        printf("Error writing is file!\n");
    }
```

مشکل کد در خط بالا رخ می دهد که در واقع در ابتدا شرط داخل حلقه اجرا می شود . تابع fputs که یک رشته را درون یک فایل قرار می دهد دارای مقادیر بازگشتی متفاوتی است . درصورتی که قرار دادن رشته در فایل با موفقیت همراه باشد ، این تابع یک مقدار نامنفی را برمی گرداند در غیر این صورت EOF را برمی گرداند . بنابراین صورت اصلاح شده قطعه کد به صورت زیر است :

```
#include <stdio.h>
int main(){

    FILE* file;
    file = fopen("output.txt","w");

    if ( file == NULL ){
        printf("could not open file!");
        return 1;
    }

    if (fputs("Hello World!\n",file) >= 0 );

    else{
        printf("Error writing is file!\n");
    }

    return 0;
}
```

b)

مشکل کد در این است که فایل دوبار باز و بسته می شود و این کار اضافی باعث می شود بهینه بودن فایل از بین برود . با استفاده از مود مناسب برای باز کردن فایل به شیوه ای که هم بتوان فایل را برای appending و هم خواندن باز کرد می توان به برنامه مطلوب رسید .

```c
#include <stdio.h>
int main(){

    FILE* file;
    file = fopen("data.txt","a+");  // changed a to a+

    if ( file == NULL ){
        printf("Could not open file!\n");
        return 1;
    }

    fprintf(file , "Newline\n");

    fseek( file , 0 , SEEK_SET);

     // moving the cursor to the beginning of the file

    char buffer[100];

    if ( fgets(buffer , sizeof(buffer) , file) == NULL ){
        printf("Error reading from file!\n");
        fclose(file);
        return 1;
    }

    else{
        printf("First line: %s\n", buffer);
    }

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

ب)

```c
#include <stdio.h>
int main(){

    FILE* file = fopen("file.txt","w");

    fputs("Height",file);
    fseek(file , 2, SEEK_SET);
    fputs("yes",file);          // Heyesight
    fseek(file , 4 , SEEK_SET);
    fputs("very",file);         // Heyeverysight
    fseek(file , 8 , SEEK_SET);
    fputs("one", file);         // Heyeveryonesight

    fseek(file , 11 , SEEK_SET);
    // cursor will move to the end of (Heyeveryone) before (sight)

    fclose(file);
    // closing file while we are at the end of (Heyeveryone)

    return 0;
}

// The output will be (Heyeveryone)
```

٢

```c
#include <stdio.h>
#define TRUE 1

int replaceNextlinesWithCommas(const char* in , const char* out){

    FILE* input = fopen(*in , "r");
    FILE* output = fopen(*out , "w");

    if ( input == NULL || output == NULL){
        fclose(input);
        fclose(output);
        return 1;
    }

    char temp;

    while (TRUE){

        temp = fgetc(input);

        if ( temp == EOF )
            break;

        if ( temp == '\n')
            temp = ',';

        if ( fputc(temp, output) == EOF ){
            fclose(input);
            fclose(output);
            return 1;
        }
    }

    fclose(input);
    fclose(output);
    return 0;
}
```

```c
#include <stdio.h>
#define MAX_LEN 30

int main(){

    FILE* Input = fopen("name.txt","r");
    FILE* Output = fopen("names_plus_grades.txt","w");

    if ( Input == NULL || Output == NULL ){
        printf("Error while opening file!");
        return 1;
    }

    char each_line[256]; // Each line(buffer) is 256 characters
    char name[MAX_LEN];
    int grade ;

    while ( fgets(each_line, sizeof(each_line), Input )){

     // reads line by line

        for ( int j = 0 ; j < MAX_LEN ; j++)
            name[j] = '\0';

        for ( int i=0 ; each_line[i] != ':' ; i++)
            name[i] = each_line[i];


        printf("%s:", name);
        scanf("%d", &grade);

        fprintf(Output,"%s:%d\n",name,grade);
    }

    fclose(Input);
    fclose(Output);
    return 0;
}
```

```c
#include <stdio.h>

int main(){

    FILE* file1 = fopen("input1.txt","r");
    FILE* file2 = fopen("input2.txt","r");
    FILE* Output = fopen("merged.txt","w");

    char line_file1[256]; // Each line is 256 characters
    char line_file2[256];

    if ( file1 == NULL || file2 == NULL || Output == NULL ){

        printf("Error while opening file(s)!");
        return 1;
    }

    int lines1 = 0 , lines2 = 0;

    while ( fgets(line_file1, sizeof(line_file1), file1)){

        lines1++ ;
    }

    while ( fgets(line_file2, sizeof(line_file2), file2)){

        lines2++ ;
    }

    fseek(file1, 0, SEEK_SET); // move the cursor to the beginning of the file.
    fseek(file2, 0, SEEK_SET);

    int count = 0;

    // to be continued in the next page
```

```
    if ( lines1 >= lines2 ){

        while ( fgets(line_file2, sizeof(line_file2), file2)){

            fgets(line_file1, sizeof(line_file1), file1);

            fprintf(Output, "%s%s",line_file2,line_file1);

            count++ ;
        }

        while ( count < lines1 ){

            fgets(line_file1, sizeof(line_file1), file1);
            fprintf(Output, "%s",line_file1);

            count++ ;
        }

    }

    else{

        while ( fgets(line_file1, sizeof(line_file1), file1)){

            fgets(line_file2, sizeof(line_file2), file2);

            fprintf(Output, "%s%s",line_file1,line_file2);

            count++ ;
        }

        while ( count < lines2 ){

            fgets(line_file2, sizeof(line_file2), file2);

            fprintf(Output, "%s",line_file2);

            count++ ;
        }

    }

    fclose(file1);
    fclose(file2);
    fclose(Output);

    return 0;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_LEN 20

int main(){

    FILE* Input = fopen("input.txt","r");
    FILE* Output = fopen("output.txt","w");

    if ( Input == NULL || Output == NULL ){

        printf("Error while opening file(s)!");
        return 1;
    }

    char* str = (char*) malloc( MAX_LEN * sizeof(char) );

    printf("Please enter your string :");
    scanf("%s",str);

    str = (char*) realloc( str , strlen(str) * sizeof(char) );


    int line = 1;
    char each_line[256];
    char separators[6] = {' ' ,',' ,'.' ,';','\n','\0'};

   // to be continued in the next page
```

```
   while ( fgets(each_line ,sizeof(each_line), Input) ){

        // reads line by line

        for ( int i = 0 ; i < sizeof(each_line) ; i++ ){

            if (each_line[i] == str[0]){

                int test = 1;
                int j,k;

                for (j=1 , k=i+1 ; j < strlen(str) ; j++ , k++){

                    if ( each_line[k] == str[j] ){

                        test++ ;
                    }

                    else break;
                }

                if ( test == strlen(str) ){

                    for ( j = 0 ; j <= 5 ; j++){

                        if ( each_line[k] == separators[j] ){

                            fprintf(Output, "%d\n", line);
                            break;
                        }
                    }
                }
            }
        }

        line++ ;
    }


    free(str);
    fclose(Input);
    fclose(Output);

    return 0;
}
```