سوال 1 بخش الف a:

```c
#include <stdio.h>
int main()
{
    FILE *file;
    file = fopen("output.txt", "w");
    if (file == NULL)
    {
        printf("Could not open file\n");
        return 1;
    }
    if (fputs("Hello, World!\n", file)==EOF )
    {
        printf("Error writing to file\n");
    }
    fclose(file);
    return 0;
}
```

سوال 1 بخش الف b :

```c
#include <stdio.h>
int main()
{
    FILE *file;
    file = fopen("data.txt", "a"); // Appending mode
    if (file == NULL)
    {
        printf("Could not open file\n");
        return 1;
    }
    fprintf(file, "New line\n");
    fclose(file);
    char buffer[100];
    file = fopen("data.txt", "r");
    if (file==NULL)//Error handeling
    {
        printf("Could not read file\n");
        return 1;
    }
    fgets(buffer, sizeof(buffer), file);
    printf("First line: %s\n", buffer);
    fclose(file);
    return 0;
}
```

سوال 1 بخش ب :

Heyeveryone

وقتی فایل را در مود write باز میکنیم و با تابع fseek پوینتر داخل فایل را جا به جا میکنیم و سپس مینویسیم مثل نوشتن در حالت insert mode vim میشود از آن ایندکسی که شروع به نوشتن میکن حرف های بعدی رو پاک میکند و جایگزین میکند و اینچنین می شود که خروجی در فایل Heyeveryone میشود

سوال 2 :

```c
#include<stdio.h>
int replaceNextlinesWithCommas(const char *in, const char *out)
{
    FILE * input=fopen(in , "r");
    FILE * output=fopen(out,"w");
    if(input==NULL||output==NULL) return 1 ;
    char ch ;
    int g=0 ;
    while(1)
    {
        ch=fgetc(input);
        if(ch==EOF) break;

        else if (ch==10)
        {
            fseek(output,g,SEEK_SET);
            fputs(".",output);

        }

        else fprintf(output ,"%c",ch) ;
        g++;

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

}
```

سوال 3 :

```c
#include <stdio.h>
#include <string.h>
int main()
{
    FILE *input = fopen("names.txt", "r");
    FILE *output = fopen("names_plus_grades.txt", "w");
    if (input == NULL || output == NULL) return 1;
    int input_num;
    char buffer[100];
    while (fgets(buffer, sizeof(buffer), input) != 0)
    {
        scanf("%d", &input_num);
        fwrite(buffer, sizeof(char), strlen(buffer), output);
        if (buffer[strlen(buffer) - 1] == '\n')
        {
            fseek(output, -2, SEEK_END);

            fprintf(output, "%d\n", input_num);
        }
        else
        {
            fseek(output, 0, SEEK_END);
            fprintf(output, "%d\0", input_num);
        }
    }
    fclose(input);
    fclose(output);
}
```

سوال 4 :

```c
#include<stdio.h>
#include<string.h>
int main()
{
    FILE *input1=fopen("input1.txt","r");
    FILE * input2=fopen("input2.txt","r");
    FILE *merged=fopen("merged.txt","w");
    if(input1==0||input2==0||merged==0) return 1 ;
    char buffer1[100] ;
    char buffer2[100];
    int flag_1=0 ;
    int flag_2=0 ;
    while(1)
    {
        if(flag_1==0&&fgets(buffer1,sizeof(buffer1),input1)==0) flag_1=1;
        if(flag_2==0&&fgets(buffer2,sizeof(buffer2),input2)==0) flag_2=1;
        if(flag_1==1&&flag_2==1) break;
        if(flag_1==0&&flag_2==0)
        {
            fputs(buffer1,merged) ;
            if(buffer1[strlen(buffer1)-1]!='\n') fputc('\n',merged);
            fputs(buffer2,merged);
            if(buffer2[strlen(buffer2)-1]!='\n') fputc('\n',merged);
            printf("1:%s\n2:%s\n",buffer1,buffer2);
        }
        else if(flag_1==0&&flag_2==1)
        {
            fputs(buffer1,merged);
            if(buffer1[strlen(buffer1)-1]!='\n') fputc('\n',merged);
            printf("1:%s\n",buffer1);
        }
        else if(flag_1==1&&flag_2==0)
        {
            fputs(buffer2,merged);
            if(buffer2[strlen(buffer2)-1]!='\n') fputc('\n',merged);
            printf("2:%s\n",buffer2);
        }
    }
    fclose(input1);
    fclose(input2);
    fclose(merged);
    return 0 ;
}
```

سوال 5 امتیازی :

```c
#include <stdio.h>
#include<string.h>
#include <stdlib.h>
void input_string(char **string, int size) {

    *string = malloc(sizeof(char) * (size));
    if (*string == NULL) {
        printf("Allocation failed\n");
        exit(1);
    }
    char ch;
    int g = 0;
    while ((ch = getchar()) != '\n' && ch != EOF) {
        if (g >= size - 1) {
            size = size + 5;
            *string = realloc(*string, size * sizeof(char));
            if (*string == NULL) {
                printf("Reallocation failed\n");
                exit(1);
            }
        }
        (*string)[g] = ch;
        (*string)[g+1]='\0';
        g++;
    }
```

```c
}
int main() {
    FILE *file = fopen("file.txt", "r");
    FILE *output = fopen("out.txt", "w");
     if (file == NULL || output == NULL) return 1 ;

    int initial_size = 10;
    char *string;
    input_string(&string, initial_size);
    char buffer[100];
    int line = 1;
    while (fgets(buffer, sizeof(buffer), file) != NULL) {
        char *word = strtok(buffer, " \n"); // tokenize the buffer
        while (word != NULL) {
            if (strcmp(word, string) == 0) {
                fprintf(output, "%d\n", line);
            }
            word = strtok(NULL, " \n");
        }
        line++;
    }
    fclose(file);
    fclose(output);
    free(string);
    return 0;
}
```