Student Name: Sahil Kakadiya

Student Number: 041052919

Course Name: C Language

a) Write a code statement that opens the file "picnic.dat" for reading purpose and assigns the returned file pointer to fPtr.

```
#include <studio.h>

Int main() {

        FILE *fptr;
        fPtr = fopen(“picnic.dat”, “r”);
        if(fptr == NULL) {
        printf(“Error opening file.\n”);
        return 1;

        }

        Fclose(fptr);
        Return 0;

}
```

a) Write a code statement that opens the file "random.dat" for writing (followed by creation) and assigns the returned file pointer to rfPtr.

```
#include <studio.h>

Int main() {

        FILE *rfptr;
        rfPtr = fopen(“random.dat”, “w”);
        if(rfptr == NULL) {
        printf(“Error opening file.\n”);
        return 1;

        }

        Fclose(rfptr);
        Return 0;

    }
```

b) Write a code statement that reads a record from the file "picnic.dat". The record that is being read consists of integer benchNum, string trailName and floating-point trailDistance. (1M)

```
#include <studio.h>

Int main() {

    FILE *fptr;
    int benchNum;
    char trailName[50];
    float trailDistance;

    fPtr = fopen(“picnic.dat”, “r”);
    if(fptr == NULL) {
    printf(“Error opening file.\n”);
    return 1;

    }

    Fscanf(fptr, “%d %s %f”, &benchNum, trailName, &trailDistance);
```

```
        Fclose(fptr);
        Return 0;

}
```

c) Write a code statement that writes a record to the file "random.dat". The record to be written consists of the integer studentID, string studentName and floating-point currentGPA.

```
#include <studio.h>

Int main() {

        FILE *rfptr;
        int studentID;
        char studentName[50];
        float currentGPA;

        rfPtr = fopen(“random.dat”, “w”);
        if(rfptr == NULL) {
        printf(“Error opening file.\n”);
        return 1;

        }

        Fscanf(rfptr, “%d %s %f”, studentID, studentName, currentGPA);

        Fclose(rfptr);
        Return 0;
}
```

d) Find the error(s) in the following program segment and write the corrected code segment

```
#include <stdio.h>

Int main() {

    FILE *fp;

    Int k;

    fp = fopen(“pizza.txt”, “r”);

    if (fp == NULL) {
        printf(“Error opening file. \n”);
        return 1;
    }


    for  (k = 0; k < 30; k++) {

        printf(“Jill likes pizza. \n”);

    }


    Fclose(fp);

    return 0;

}
```

e) Suppose we have these statements in a program (3M)

```
FILE *fp1, *fp2; char ch;
fp1 = fopen ("file1", "r");
fp2 = fopen ("file2", "w");
```

Assuming that both files opened successfully, supply the missing arguments in the following function calls:

```
1) fscanf (fp1, "%c", &ch);
```

2) fprintf (fp2, "%c\n", ch);

3) fclose (fp2); /* close the file for writing */