

```

1 // GROUP B
2 // Benjamin Welch
3 // beb.welch@okstate.edu
4
5
6 #include "header.h"
7
8 int GetTomorrowDate(char *tomorrowsDate)
9 {
10     FILE *fp;
11
12
13     // variables to store the date and time components
14     int hours, minutes, seconds, day, month, year;
15
16     // `time_t` is an arithmetic time type
17     time_t now = (int)time(NULL) + 86400;
18
19     // localtime converts a `time_t` value to calendar time and
20     // returns a pointer to a `tm` structure with its members
21     // filled with the corresponding values
22     struct tm *local = localtime(&now);
23
24     hours = local->tm_hour;    // get hours since midnight (0-23)
25     minutes = local->tm_min;    // get minutes passed after the hour (0-59)
26     seconds = local->tm_sec;    // get seconds passed after a minute (0-59)
27
28     day = local->tm_mday;    // get day of month (1 to 31)
29     month = local->tm_mon + 1;    // get month of year (0 to 11)
30     year = local->tm_year + 1900;    // get year since 1900
31
32     fp = fopen("dates.txt", "w+");
33
34     // print the current date
35     fprintf(fp, "%02d/%02d/%d", month, day, year);
36     fclose(fp);
37
38     fp = fopen("dates.txt", "r");
39
40     fgets(tomorrowsDate, 11, fp);
41     // printf("%s", tomorrowsDate);
42     fclose(fp);
43
44
45     return 0;
46 }
47
48 int GetTodayDate(char *todaysDate)
49 {
50     FILE *fp;
51
52     // variables to store the date and time components
53     int hours, minutes, seconds, day, month, year;
54
55     // `time_t` is an arithmetic time type
56     time_t now = (int)time(NULL);
57
58     // localtime converts a `time_t` value to calendar time and
59     // returns a pointer to a `tm` structure with its members
60     // filled with the corresponding values
61     struct tm *local = localtime(&now);
62
63
64     day = local->tm_mday;    // get day of month (1 to 31)
65     month = local->tm_mon + 1;    // get month of year (0 to 11)
66     year = local->tm_year + 1900;    // get year since 1900
67

```

```

68 // print the current date
69 fp = fopen("dates.txt", "w+");
70
71 // print the current date
72 fprintf(fp, "%02d/%02d/%d", month, day, year);
73 fclose(fp);
74
75 fp = fopen("dates.txt", "r");
76
77 fgets(todaysDate, 11, fp);
78 // printf("%s", todaysDate);
79 fclose(fp);
80
81 return 0;
82 }
83
84 int seatChecker(int trainNum)
85 {
86     FILE *fp ;
87     char c;
88     int zeros = 0;
89
90
91     printf( "Opening the file train in read mode \n" );
92     if (trainNum == 1)
93     {
94         fp = fopen ( "train1.txt", "r" ); // opening an existing file
95     }
96     else if (trainNum == 2)
97     {
98         fp = fopen ( "train2.txt", "r" ); // opening an existing file
99     }
100     if ( fp == NULL )
101     {
102         printf ( "Could not open file train \n" );
103         return 1;
104     }
105     printf( "Reading the file train \n" );
106     while ( 1 )
107     {
108         c = fgetc ( fp ); // reading the file
109
110         if(c == '0')
111         {
112             zeros++;
113             printf("%0d\n", zeros);
114         }
115
116         if ( c == EOF )
117             break ;
118         //printf ( "%c", c );
119     }
120     printf("Closing the file train \n");
121     fclose ( fp ); // Closing the file
122     return zeros;
123 }

```