```c
// GROUP B
// Benjamin Welch
// ben.welch@okstate.edu


#include "header.h"


// A function to get calculate tomorrows date and return it for use in checking the correct date by the client
int GetTomorrowDate(char *tomorrowsDate)
{
    FILE *fp;


    // variables to store the date and time components
    int hours, minutes, seconds, day, month, year;

    // `time_t` is an arithmetic time type
    time_t now = (int)time(NULL) + 86400;

    // localtime converts a `time_t` value to calendar time and
    // returns a pointer to a `tm` structure with its members
    // filled with the corresponding values
    struct tm *local = localtime(&now);

    day = local->tm_mday;            // get day of month (1 to 31)
    month = local->tm_mon + 1;       // get month of year (0 to 11)
    year = local->tm_year + 1900;    // get year since 1900

    fp = fopen("dates.txt", "w+");

    // print the current date
    fprintf(fp, "%02d/%02d/%d", month, day, year);
    fclose(fp);

    fp = fopen("dates.txt", "r");

    fgets(tomorrowsDate, 11, fp);
    fclose(fp);


    return 0;
}


// A function to get calculate todays date and return it for use in checking the correct date by the client
int GetTodayDate(char *todaysDate)
{
    FILE *fp;

    // variables to store the date and time components
    int hours, minutes, seconds, day, month, year;

    // `time_t` is an arithmetic time type
    time_t now = (int)time(NULL);

    // localtime converts a `time_t` value to calendar time and
    // returns a pointer to a `tm` structure with its members
    // filled with the corresponding values
    struct tm *local = localtime(&now);


    day = local->tm_mday;            // get day of month (1 to 31)
    month = local->tm_mon + 1;       // get month of year (0 to 11)
    year = local->tm_year + 1900;    // get year since 1900

    // print the current date
```

```
68      fp = fopen("dates.txt", "w+");
69
70      // print the current date
71      fprintf(fp, "%02d/%02d/%d", month, day, year);
72      fclose(fp);
73
74      fp = fopen("dates.txt", "r");
75
76      fgets(todaysDate, 11, fp);
77      fclose(fp);
78
79      return 0;
80  }
81
82
83  // This function is to check through the selected train file and count how many seats are not taken
84  // and then return that value to the server.
85  int seatChecker(int trainNum)
86  {
87     FILE *fp ;
88     char c;
89     int zeros = 0;
90
91
92     printf( "Opening the file train in read mode \n" ) ;
93     if (trainNum == 1)
94     {
95        fp = fopen ( "train1.txt", "r" ) ; // opening an existing file
96     }
97     else if (trainNum == 2)
98     {
99        fp = fopen ( "train2.txt", "r" ) ; // opening an existing file
100    }
101    if ( fp == NULL )
102    {
103      printf ( "Could not open file train \n" ) ;
104      return 1;
105    }
106    printf( "Reading the file train \n" ) ;
107    while ( 1 )
108    {
109      c = fgetc ( fp ) ; // reading the file
110
111      if(c == '0')
112      {
113         zeros++;
114         printf("%d\n", zeros);
115      }
116
117      if ( c == EOF )
118      break ;
119      //printf ( "%c", c );
120    }
121    printf("Closing the file train \n") ;
122    fclose ( fp ) ; // Closing the file
123    return zeros;
124 }
```