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


#include "header.h"

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);

    hours = local->tm_hour;         // get hours since midnight (0-23)
    minutes = local->tm_min;        // get minutes passed after the hour (0-59)
    seconds = local->tm_sec;        // get seconds passed after a minute (0-59)

    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);
    // printf("%s", tomorrowsDate);
    fclose(fp);


    return 0;
}

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

```

```c
    // print the current date
    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(todaysDate, 11, fp);
    // printf("%s", todaysDate);
    fclose(fp);

    return 0;
}

int seatChecker(int trainNum)
{
   FILE *fp ;
   char c;
   int zeros = 0;


   printf( "Opening the file train in read mode \n" ) ;
   if (trainNum == 1)
   {
      fp = fopen ( "train1.txt", "r" ) ; // opening an existing file
   }
   else if (trainNum == 2)
   {
      fp = fopen ( "train2.txt", "r" ) ; // opening an existing file
   }
   if ( fp == NULL )
   {
     printf ( "Could not open file train \n" ) ;
     return 1;
   }
   printf( "Reading the file train \n" ) ;
   while ( 1 )
   {
     c = fgetc ( fp ) ; // reading the file

     if(c == '0')
     {
        zeros++;
        printf("%d\n", zeros);
     }

     if ( c == EOF )
     break ;
     //printf ( "%c", c );
   }
   printf("Closing the file train \n") ;
   fclose ( fp ) ; // Closing the file
   return zeros;
}
```