Assignment HW7
Due date (on or before): Announced in class.

1. Textbook, page 246, Exercise 6.21 (5 points)

6.21 *(Airline Reservations System)* A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You'll write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats).

Your program should display the following menu of alternatives:

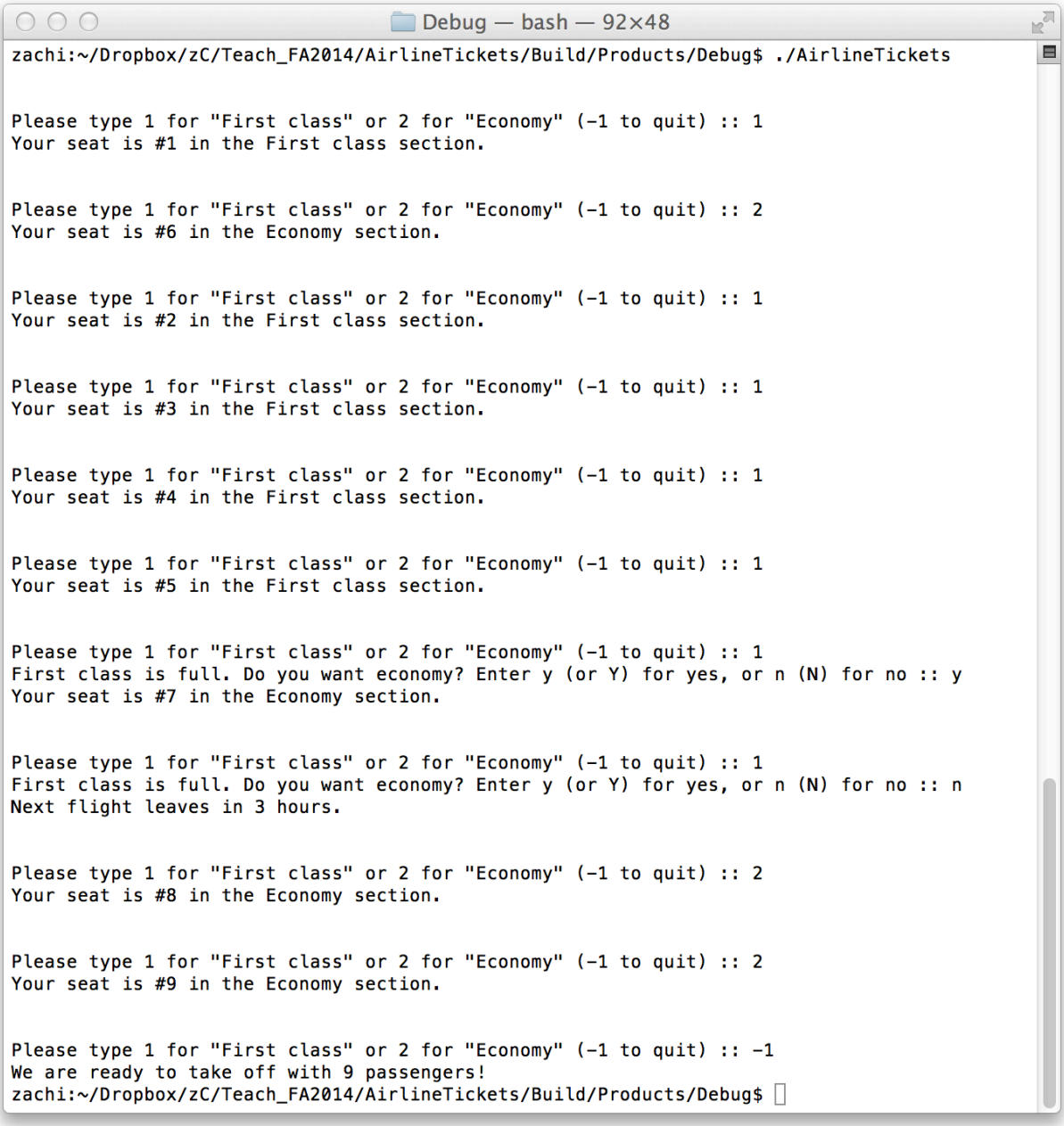
```
Please type 1 for "first class"
Please type 2 for "economy"
```

If the person types 1, then your program should assign a seat in the first class section (seats 1–5). If the person types 2, then your program should assign a seat in the economy section (seats 6–10). Your program should then print a boarding pass indicating the person's seat number and whether it's in the first class or economy section of the plane.

Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding element of the array to 1 to indicate that the seat is no longer available.

Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, your program should ask the person if it's acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message `"Next flight leaves in 3 hours."`

Screen shot below:



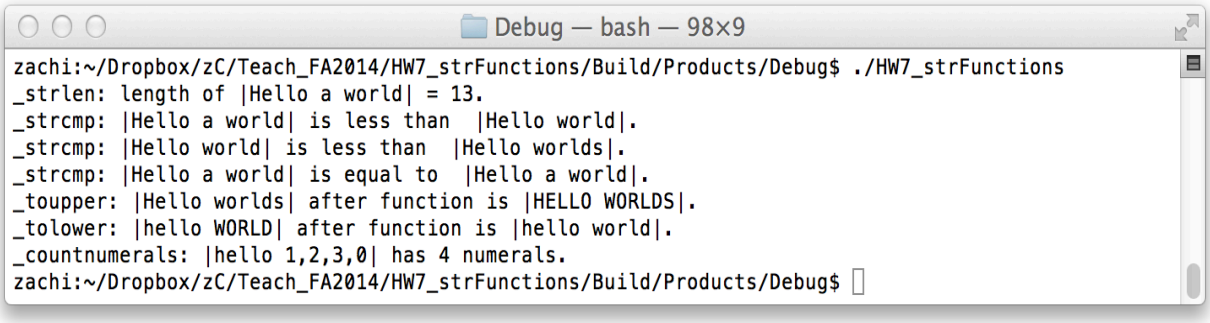
********** Start of code + Screen shot**********

********** End of code + Screen shot**********

2. String operations

Create a new project, and copy the below code into your main.c file.
Implement the below functions declared in the skeleton code.

The results should look like the below screenshot (please attach yours!!)




```
/**** Start of code to complete ****/
//
//  main.c
//  HW7_strFunctions
//
//

#include <stdio.h>

// _strlen: length of a string
int _strlen(char s[]);

// _strcmp: returns negative, zero, or positive if s is
less, equal, or greater than t (lexicographicaly)
int _strcmp(char s[], char t[]);

// _toupper: returns the string with all lower cases
converted to upper, and as int the number of changes
void _toupper(char s[]);

// _tolower: you guess...
void _tolower(char s[]);

// _countnumerals: returns how many numerals are in the
```

```
string.
int _countnumerals(char s[]);

//
// testing code!!
// Don't change anything in the below code.
char* printcomp(int n)
{
    char* pstr=NULL;
    if ( n==0 )
        pstr = "equals to";
    if ( n<0 )
        pstr = "less than";
    if ( n >0 )
        pstr = "greater than";

    return (pstr);
}

int main()
{

    char str0[80]= "Hello world";
    char str1[80]= "Hello a world";
    char str2[80]= "Hello worlds";
    char str3[80]= "hello WORLD";
    char str4[80]= "hello 1,2,3,0";

    printf("_strlen: length of |%s| =
%d.\n",str1,_strlen(str1));


    printf("_strcmp: |%s| is %s  |%s|.\n",str1,
printcomp(_strcmp(str1, str0)) ,str0);
    printf("_strcmp: |%s| is %s  |%s|.\n",str0,
printcomp(_strcmp(str0, str2)) ,str2);
    printf("_strcmp: |%s| is %s  |%s|.\n",str1,
printcomp(_strcmp(str1, str1)) ,str1);
```

```
    printf("_toupper: |%s| after function is ",str2);
    _toupper(str2);
    printf("|%s|.\n",str2);

    printf("_tolower: |%s| after function is ",str3);
    _tolower(str3);
    printf("|%s|.\n",str3);


    printf("_countnumerals: |%s| has %d
numerals.\n",str4,_countnumerals(str4));


    return 0;
}
/**** End of code to complete ****/
```

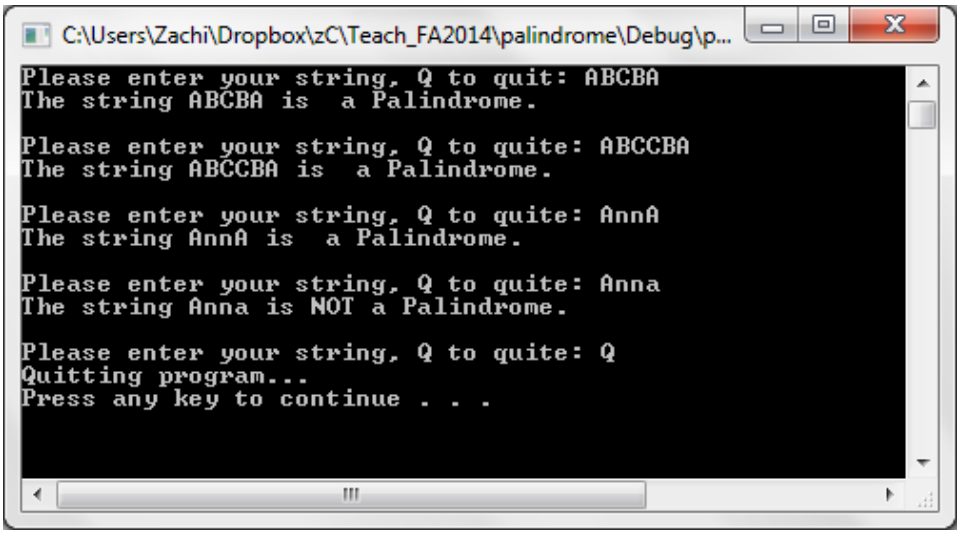
********** Start of code + Screen shot**********

********** End of code + Screen shot**********

3. Palindrome
(similar to Textbook, page 251, Exercise 6.31)

Write a program that determines whether a string is a palindrome. See example screenshot below.
The program will ask the user for an input string (assume not longer than 79 characters). The program will then print whether the string is a palindrome or not, and prompt the user to input new string. The process will stop when the user inputs a string of one character, Q (if entered at the first time, no palindrome is checked).

What is a palindrome? A string that spells the same forward and backward.
Example:
ABCBA is a palindrome
ABCCBA is a palindrome
AnnA is a palindrome
ABCD is NOT a palindrome
Anna is NOT a palindrome (we are case sensitive here)



********** Start of code + Screen shot**********

********** End of code + Screen shot**********