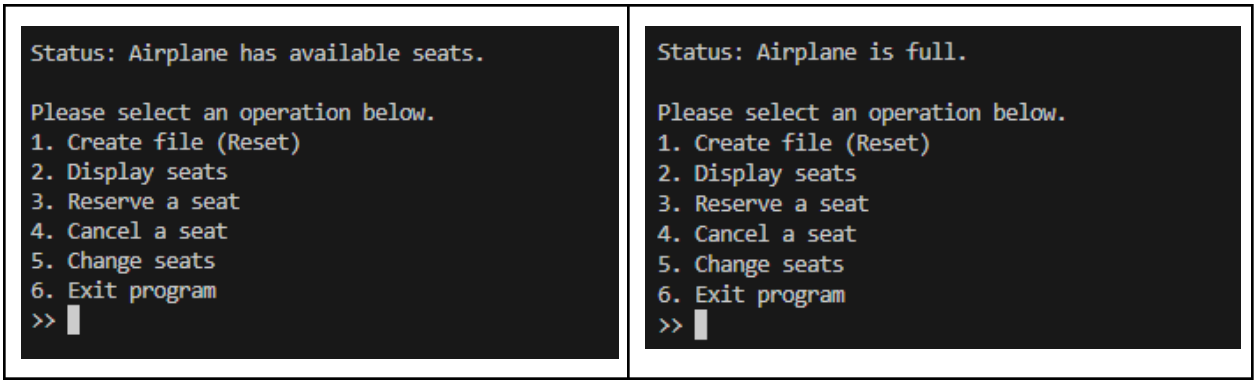
Cabral, Shikina Y.
BSCS 1-5 | Computer Programming 2

PROJECT: Airplane Seating Program in C

OBJECTIVE: An Airplane seating program that could accommodate 40 passengers, having 10 rows and 4 columns. It must satisfy the following requirement: (A) Seat reservation, (B) Change seats, (C) Cancel selected seat, (D) Display available seat.

COMPILING: The compiler used in the making of this program is Visual Studio Code (VS Code). The file name for the file pointer is “seats.txt”

PROCESS:
Upon compiling and running, the status of the seats’ availability will be shown at the uppermost section of the screen. See figure below,



The user will be prompted to choose among 6 operations:

1. Create file (reset): This will reset the file (plane.txt), overwriting the contents of it. After prompting this operation, all the seats will be set to unoccupied.

2. Display seats: This will display the availability of the seats. If the seats are full, a comment “Airplane is full” will be printed at the bottom section of the program.

3. Reserve a seat: This will prompt the user to select a seat. If the selected seat is unavailable, the program will prompt the user to select another seat.

4. Change seats: This will change the selected seat of the user. The program will prompt the user to type-in their current seat then make them select a new seat as a replacement for the previous seat. If the given current seat is not yet reserved, the user will choose whether to continue with the reservation or not.

5. Cancel a seat: This will free up the selected seat. If the selected seat is not yet reserved, the program will print “No one has reserved this seat yet.”.

6. Exit program: This will terminate the program.

All operations done within the whole duration of the program will be saved in the file ‘seats.txt’.

<table>
<tr><th colspan="2">Global Declaration</th></tr>
<tr><td>

```c
#include<stdio.h>
#include<stdbool.h>
#include<stdlib.h>
#include<ctype.h>
#include<conio.h>
#define ROW 10
#define COL 4

struct plane {
    int row;
    char col;
    bool occupied;
}seat[ROW][COL];

typedef struct plane plane;
```

</td><td>In this program, both the row and column are declared globally to avoid repeated declarations.

The variables inside the struct plane include the row, col, and the seat’s availability in boolean format. 2D array is also declared globally.

I used the typedef function to shorten the ‘struct plane’ into simply ‘plane’.</td></tr>
<tr><th colspan="2">In charToInt()</th></tr>
<tr><td>

```c
int charToInt(char a){
    return a - 'A';
}
```

</td><td>This function will return the int equivalent of the char.

Ex. B = 1
B - ‘A’ (the ASCII equivalent of A is 65 and 66 for B)
66 - 65 = 1

This is used to convert the user input for ‘column’ to an integer. Useful to access the array.</td></tr>
<tr><th colspan="2">In readFile()</th></tr>
<tr><td>

```c
void readFile(FILE *fp){
    char n;
```

</td><td>This function aims to read the contents of the file.</td></tr>
</table>

```
    for(int i=0; i<ROW; i++){
        fscanf(fp, "%d", &seat[i][0].row);
        for (int j=0; j<COL; j++){
            fscanf(fp, " %c", &n);
            if (n == 'X'){
                seat[i][j].occupied = true;
            }
            else {
                seat[i][j].occupied = false;
            }
        }
        fgetc(fp);
    }
}
```

fscanf() is used to read from the file. If the fscanf() receives the letter 'X', the pointed seat will be set to occupied.

fgetc() is used to get the newline buffer.

In writeFile()

```
void writeFile(FILE *fp){
    fseek(fp, 0, SEEK_SET);
    for (int i=0; i<ROW; i++){
        fprintf(fp, "%d\t", i+1);
        for (int j=0; j<COL; j++){
            seat[i][j].row = i + 1;
            seat[i][j].col = j + 'A';
            if (seat[i][j].occupied){
                fprintf(fp, "X\t");
            }
            else {
                fprintf(fp, "%c\t",
seat[i][j].col);
            }
        }
        fprintf(fp, "\n");
    }
}
```

This function will write to the file.

Here I used the fseek() function to place the file pointer at the beginning of the file.

Then I used the fprinf() function to write to file (per row). In the inner loop I initialized the contents of the seat array.

In isFull()

```
int isFull(){                          // Return 1
if seats are full
    for (int i=0; i<ROW; i++){
```

This function will return 1 if all the seats are full and 0 if there are still remaining seats left.

```
        for (int j=0; j<COL; j++){
            if (seat[i][j].occupied == false)
return 0;
        }
    }
    return 1;
}
```

In createFile()

```
void createFile(){
    system("cls");

    FILE *fp = fopen("seats.txt", "w");

    // Initialize and print seats
    for (int i=0; i<ROW; i++){
        fprintf(fp, "%d\t", i+1);
        for (int j=0; j<COL; j++){
            seat[i][j].row = i + 1;
            seat[i][j].col = j + 'A';
            seat[i][j].occupied = false;
            fprintf(fp, "%c\t", seat[i][j].col);
        }
        fprintf(fp, "\n");
    }

    fclose(fp);
}
```

This function will create another file under the same file name; Or simply resets the content of the file.

The file ‘seats.txt’ will be opened in write mode.

This function will also initialize the seats and set all the .occupied booleans to false. As the initialization occurs, seats are also being printed in the file ‘seats.txt’.

In displaySeat()

```
void displaySeat(){
    system("cls");
    FILE *fp = fopen("seats.txt", "r");

    // Prints content of file (per line)
    char text[100];
    while (fgets(text, sizeof(text), fp)){
        printf("%s", text);
    }
```

This will display the seats’ availability.

Here the file will be opened in ‘read’ mode. Then all contents will be printed on the screen.

At the end of the function, the isFull() function is called. If the seats are full, it will print “Airplane is full”.

```
    if (isFull()){
        printf("\nAirplane is full.");
    }

    printf("\n\nPress any key to continue...");
    getch();
}
```

In reserveSeat()

```
void reserveSeat(){
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    if (isFull() == 0){
        int rowTemp, colTemp2;
        char colTemp;
        printf("Enter desired seat: ");
        scanf("%d%c", &rowTemp, &colTemp);
        colTemp2 = charToInt(toupper(colTemp));

        while
(seat[rowTemp-1][colTemp2].occupied == true){
            printf("\nSeat %d%c is already
occupied. Please choose another: ", rowTemp,
colTemp);
            scanf("%d%c", &rowTemp, &colTemp);
            colTemp2 =
charToInt(toupper(colTemp));
        }

        seat[rowTemp-1][colTemp2].occupied =
true;
        printf("Seat %d%c successfully booked.",
rowTemp, colTemp);
    }
```

This function will assign seats according to the inputted seat of the user.

File ‘seats.txt’ will be opened in ‘r+’ mode to read then write.

readFile() is called to fetch the contents of the file.

Here the isFull() function is called which will only allow users to reserve seats if the plane is not yet full; otherwise it will print “Airplane is full”.

If the inputted seat is already occupied, the user will be prompted to choose another seat. This will loop until the user has selected an unoccupied seat.

toupper() function is used to handle the capitalization of the user input because ‘a’ and ‘A’ have different ASCII equivalents (important for the charToInt() function).

After the seat is reserved, writeFile() function will be called to save the seat.

```
    else {printf("Airplane is full.");}

    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}
```

In changeSeat()

```
void changeSeat(){
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    if (isFull() == 0){
        int rowTemp, rowTemp2, colTemp1,
colTemp2;
        char colTemp;
        printf("Enter current seat: ");
        scanf("%d%c", &rowTemp, &colTemp);
        colTemp1 = charToInt(toupper(colTemp));

        if(seat[rowTemp-1][colTemp1].occupied){
            printf("Enter desired seat: ");
            scanf("%d%c", &rowTemp2, &colTemp);
            colTemp2 =
charToInt(toupper(colTemp));


while(seat[rowTemp2-1][colTemp2].occupied){
                printf("\nDesired seat is
occupied. Please choose another: ");
                scanf("%d%c", &rowTemp2,
&colTemp);
                colTemp2 =
charToInt(toupper(colTemp));
            }
```

This function is for changing seats.

The file is opened in r+ mode to allow seeking, reading, and writing to file.

readFile() is called to fetch the contents of the file.

Here the isFull() function is called which will only allow users to change seats if the plane is not yet full; otherwise it will print “Airplane is full”.

The user will be prompted to enter their current seat and (1) if that seat is occupied, they will be prompted to enter their desired seat:

a. If desired seat is occupied, the user will be prompted to select another seat (loop until desired seat is available)
b. If desired seat is available, their current seat will be changed to their desired seat and the previous seat will be set to unoccupied.

However (2) if the current seat that they entered is unoccupied, they will be prompted to choose whether to continue with the reservation or not.

At the end of the code, writeFile() will be called to save the changes done.

```
            seat[rowTemp-1][colTemp1].occupied =
false;
            seat[rowTemp2-1][colTemp2].occupied
= true;
            printf("Seat %d%c confirmed.",
rowTemp2, colTemp);
        }

        else {
            printf("\nNo one has booked %d%c
seat yet. Press 1 to continue, 0 to cancel.\n",
rowTemp, colTemp);
            printf(">> ");
            int choice;
            scanf("%d", &choice);

            switch (choice){
                case 1:
                    printf("Enter desired seat:
");
                    scanf("%d%c", &rowTemp2,
&colTemp);
                    colTemp2 =
charToInt(toupper(colTemp));


while(seat[rowTemp2-1][colTemp2].occupied){
                        printf("\nDesired seat
is occupied. Please choose another: ");
                        scanf("%d%c", &rowTemp2,
&colTemp);
                        colTemp2 =
charToInt(toupper(colTemp));
                    }


seat[rowTemp2-1][colTemp2].occupied = true;
                    printf("Seat %d%c
confirmed.", rowTemp2, colTemp);
```

```
                    break;

                case 0:
                    break;
            }
        }
    }

    else {printf("Airplane is full.");}

    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}
```

In cancelSeat()

```
void cancelSeat(){                    /
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    int rowTemp, colTemp2;
    char colTemp;
    printf("Enter which seat to cancel: ");
    scanf("%d%c", &rowTemp, &colTemp);
    colTemp2 = charToInt(toupper(colTemp));

    if (seat[rowTemp-1][colTemp2].occupied ==
false){
        printf("\nNo one has reserved this seat
yet.");
    }

    else {
        seat[rowTemp-1][colTemp2].occupied =
false;
        printf("\nSeat %d%c cancelled.",
```

This function will free up the seat entered by the user.

readFile() is called to fetch the contents of the file.

The user will be prompted to enter a seat they want to cancel,

a. If the seat is occupied, the program will free up the seat.
b. If the seat is not occupied, the program will print “No one has reserved this seat yet”.

The writeFile() function will be called to save all the changes.

```
rowTemp, colTemp);
    }

    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}
```

In main()

```
int main(){
    int choice;

    FILE *fp = fopen("seats.txt", "r");
    if (fp == NULL){        // If file does not
exist, createFile() is called
        createFile();
    }

    do {
        system("cls");
        if (isFull()) {printf("Status: Airplane
is full.\n\n");}
        else {printf("Status: Airplane has
available seats.\n\n");}
        printf("Please select an operation
below.\n");
        printf("1. Create file (Reset)\n");
        printf("2. Display seats\n");
        printf("3. Reserve a seat\n");
        printf("4. Cancel a seat\n");
        printf("5. Change seats\n");
        printf("6. Exit program\n");
        printf(">> ");
        scanf("%d", &choice);

        switch(choice){
            case 1:
                createFile();
```

File pointer *fp is created to open the file ‘seats.txt’ in read mode. Then, I implemented an error handling that will work if no ‘seats.txt’ exists in the system. If the execution falls on that error handling, the createFile() function will be called to create the file.

A do-while loop is created to handle the operations of the program. At the start of the loop, an if-else function is used to check if the seats are full. This conditional statement will call the isFull() function and will print the status of the plane.

A switch loop is used to redirect the user to the appropriate function based on their input.

```
                printf("File created. You may
proceed to reservation now.");
                printf("\n\nPress any key to
continue...");
                getch();
                break;
            case 2:
                displaySeat();
                break;
            case 3:
                reserveSeat();
                break;
            case 4:
                cancelSeat();
                break;
            case 5:
                changeSeat();
                break;
            case 6:
                break;
            default:
                printf("\nInvalid input.");
                printf("\n\n Press any key to
continue...");
                getch();
        }
    }while (choice != 6);
    printf("\nProgram terminated.");
}
```

Complete Code:

```c
#include<stdio.h>
#include<stdbool.h>
#include<stdlib.h>
#include<ctype.h>
#include<conio.h>
#define ROW 10
#define COL 4

struct plane {
    int row;
    char col;
    bool occupied;
}seat[ROW][COL];

typedef struct plane plane;

int charToInt(char a){              // Converts char to int
    return a - 'A';
}

void readFile(FILE *fp){            // Reads from file
    char n;
    for(int i=0; i<ROW; i++){
        fscanf(fp, "%d", &seat[i][0].row);
        for (int j=0; j<COL; j++){
            fscanf(fp, " %c", &n);
            if (n == 'X'){
                seat[i][j].occupied = true;
            }
            else {
                seat[i][j].occupied = false;
            }
        }
        fgetc(fp);
    }
}

void writeFile(FILE *fp){           // Prints changes to file
```

```
    fseek(fp, 0, SEEK_SET);
    for (int i=0; i<ROW; i++){
        fprintf(fp, "%d\t", i+1);
        for (int j=0; j<COL; j++){
            seat[i][j].row = i + 1;
            seat[i][j].col = j + 'A';
            if (seat[i][j].occupied){
                fprintf(fp, "X\t");
            }
            else {
                fprintf(fp, "%c\t", seat[i][j].col);
            }
        }
        fprintf(fp, "\n");
    }
}

int isFull(){                       // Return 1 if seats are full
    for (int i=0; i<ROW; i++){
        for (int j=0; j<COL; j++){
            if (seat[i][j].occupied == false) return 0;
        }
    }
    return 1;
}

// Program Operations
void createFile(){                  // Creates file and initializes seats
    system("cls");

    FILE *fp = fopen("seats.txt", "w");

    // Initialize and print seats
    for (int i=0; i<ROW; i++){
        fprintf(fp, "%d\t", i+1);
        for (int j=0; j<COL; j++){
            seat[i][j].row = i + 1;
            seat[i][j].col = j + 'A';
            seat[i][j].occupied = false;
```

```
            fprintf(fp, "%c\t", seat[i][j].col);
        }
        fprintf(fp, "\n");
    }

    fclose(fp);
}

void displaySeat(){                 // Displays file
    system("cls");
    FILE *fp = fopen("seats.txt", "r");

    // Prints content of file (per line)
    char text[100];
    while (fgets(text, sizeof(text), fp)){
        printf("%s", text);
    }

    if (isFull()){
        printf("\nAirplane is full.");
    }

    printf("\n\nPress any key to continue...");
    getch();
}

void reserveSeat(){                 // Assigns seat
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    if (isFull() == 0){
        int rowTemp, colTemp2;
        char colTemp;
        printf("Enter desired seat: ");
        scanf("%d%c", &rowTemp, &colTemp);
        colTemp2 = charToInt(toupper(colTemp));
```

```
        while (seat[rowTemp-1][colTemp2].occupied == true){
            printf("\nSeat %d%c is already occupied. Please choose another: ",
rowTemp, colTemp);
            scanf("%d%c", &rowTemp, &colTemp);
            colTemp2 = charToInt(toupper(colTemp));
        }

        seat[rowTemp-1][colTemp2].occupied = true;
        printf("Seat %d%c successfully booked.", rowTemp, colTemp);
    }

    else {printf("Airplane is full.");}

    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}

void cancelSeat(){                  // Cancels seat
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    int rowTemp, colTemp2;
    char colTemp;
    printf("Enter which seat to cancel: ");
    scanf("%d%c", &rowTemp, &colTemp);
    colTemp2 = charToInt(toupper(colTemp));

    if (seat[rowTemp-1][colTemp2].occupied == false){
        printf("\nNo one has reserved this seat yet.");
    }

    else {
        seat[rowTemp-1][colTemp2].occupied = false;
        printf("\nSeat %d%c cancelled.", rowTemp, colTemp);
    }
```

```
    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}

void changeSeat(){                  // Changes seat
    system("cls");

    FILE *fp = fopen("seats.txt", "r+");
    readFile(fp);

    if (isFull() == 0){
        int rowTemp, rowTemp2, colTemp1, colTemp2;
        char colTemp;
        printf("Enter current seat: ");
        scanf("%d%c", &rowTemp, &colTemp);
        colTemp1 = charToInt(toupper(colTemp));

        if(seat[rowTemp-1][colTemp1].occupied){
            printf("Enter desired seat: ");
            scanf("%d%c", &rowTemp2, &colTemp);
            colTemp2 = charToInt(toupper(colTemp));

            while(seat[rowTemp2-1][colTemp2].occupied){
                printf("\nDesired seat is occupied. Please choose another: ");
                scanf("%d%c", &rowTemp2, &colTemp);
                colTemp2 = charToInt(toupper(colTemp));
            }

            seat[rowTemp-1][colTemp1].occupied = false;
            seat[rowTemp2-1][colTemp2].occupied = true;
            printf("Seat %d%c confirmed.", rowTemp2, colTemp);
        }

        else {
            printf("\nNo one has booked %d%c seat yet. Press 1 to continue, 0 to
cancel.\n", rowTemp, colTemp);
```

```
            printf(">> ");
            int choice;
            scanf("%d", &choice);

            switch (choice){
                case 1:
                    printf("Enter desired seat: ");
                    scanf("%d%c", &rowTemp2, &colTemp);
                    colTemp2 = charToInt(toupper(colTemp));

                    while(seat[rowTemp2-1][colTemp2].occupied){
                        printf("\nDesired seat is occupied. Please choose another:
");
                        scanf("%d%c", &rowTemp2, &colTemp);
                        colTemp2 = charToInt(toupper(colTemp));
                    }

                    seat[rowTemp2-1][colTemp2].occupied = true;
                    printf("Seat %d%c confirmed.", rowTemp2, colTemp);
                    break;

                case 0:
                    break;
            }
        }
    }

    else {printf("Airplane is full.");}

    writeFile(fp);
    fclose(fp);
    printf("\n\nPress any key to continue...");
    getch();
}

int main(){
    int choice;

    FILE *fp = fopen("seats.txt", "r");
```

```c
    if (fp == NULL){        // If file does not exist, createFile() is called
        createFile();
    }

    do {
        system("cls");
        if (isFull()) {printf("Status: Airplane is full.\n\n");}
        else {printf("Status: Airplane has available seats.\n\n");}
        printf("Please select an operation below.\n");
        printf("1. Create file (Reset)\n");
        printf("2. Display seats\n");
        printf("3. Reserve a seat\n");
        printf("4. Cancel a seat\n");
        printf("5. Change seats\n");
        printf("6. Exit program\n");
        printf(">> ");
        scanf("%d", &choice);

        switch(choice){
            case 1:
                createFile();
                printf("File created. You may proceed to reservation now.");
                printf("\n\nPress any key to continue...");
                getch();
                break;
            case 2:
                displaySeat();
                break;
            case 3:
                reserveSeat();
                break;
            case 4:
                cancelSeat();
                break;
            case 5:
                changeSeat();
                break;
            case 6:
                break;
```

```
            default:
                printf("\nInvalid input.");
                printf("\n\n Press any key to continue...");
                getch();
        }
    }while (choice != 6);
    printf("\nProgram terminated.");
    fclose(fp);
}
```

Sample Run:

a. On execution

```
Status: Airplane has available seats.

Please select an operation below.
1. Create file (Reset)
2. Display seats
3. Reserve a seat
4. Cancel a seat
5. Change seats
6. Exit program
>>
```

b. Reserve 10D seat

```
Enter desired seat: 10d
Seat 10d successfully booked.

Press any key to continue...
```

c. Display seats

```
1       A       B       C       D
2       A       B       C       D
3       A       B       C       D
4       A       B       C       D
5       A       B       C       D
6       A       B       C       D
7       A       B       C       D
8       A       B       C       D
9       A       B       C       D
10      A       B       C       X


Press any key to continue...
```

d. Change seat from 10D to 4B

```
Enter current seat: 10d
Enter desired seat: 4b
Seat 4b confirmed.

Press any key to continue...
```

e. Cancel 4B seat

```
Enter which seat to cancel: 4b

Seat 4b cancelled.

Press any key to continue...
```

f. Reserve 1A seat then terminate the program

seat.c | seats.txt

output > seats.txt

```
1   1   X   B   C   D
2   2   A   B   C   D
3   3   A   B   C   D
4   4   A   B   C   D
5   5   A   B   C   D
6   6   A   B   C   D
7   7   A   B   C   D
8   8   A   B   C   D
9   9   A   B   C   D
10  10  A   B   C   D
11
```