Design a flight seat allocation system using arrays. Your system should allow passengers to book seats, cancel bookings, and view the current seating arrangements. Include features to display available seats and handle invalid inputs gracefully.

```
#include <stdio.h>
#define ROWS 5
#define COLUMNS 4

void displaySeats(char seats[ROWS][COLUMNS]) {
    printf("\nSeating Arrangement:\n");
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            printf("%c ", seats[i][j]);
        }
        printf("\n");
    }
}

void displayAvailableSeats(char seats[ROWS][COLUMNS]) {
    printf("\nAvailable Seats:\n");
    int count = 0;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            if (seats[i][j] == 'O') {
                printf("Row %d, Column %d\n", i + 1, j + 1);
                count++;
            }
        }
    }
    if (count == 0) {
        printf("No seats available!\n");
    }
}

void bookSeat(char seats[ROWS][COLUMNS]) {
    int row, col;
    printf("Enter row (1-%d) and column (1-%d) to book a seat: ", ROWS, COLUMNS);
    scanf("%d %d", &row, &col);

    if (row < 1 || row > ROWS || col < 1 || col > COLUMNS) {
        printf("Invalid seat! Please select a valid seat.\n");
        return;
    }

    if (seats[row - 1][col - 1] == 'X') {
        printf("Seat already booked! Try a different seat.\n");
    } else {
        seats[row - 1][col - 1] = 'X';
        printf("Seat booked successfully.\n");
    }
}

void cancelBooking(char seats[ROWS][COLUMNS]) {
    int row, col;
    printf("Enter row (1-%d) and column (1-%d) to cancel booking: ", ROWS, COLUMNS);
    scanf("%d %d", &row, &col);

    if (row < 1 || row > ROWS || col < 1 || col > COLUMNS) {
        printf("Invalid seat! Please select a valid seat.\n");
        return;
    }

    if (seats[row - 1][col - 1] == 'O') {
        printf("Seat is not booked! Nothing to cancel.\n");
    } else {
        seats[row - 1][col - 1] = 'O';
        printf("Booking canceled successfully.\n");
    }
}

int main() {
    char seats[ROWS][COLUMNS];
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            seats[i][j] = 'O'; // Initialize all seats as available ('O')
        }
    }

    int choice;
    for(;;)
    {
        printf("\nFlight Seat Allocation System\n");
        printf("1. View Current Seating Arrangement\n");
        printf("2. Display Available Seats\n");
        printf("3. Book a Seat\n");
        printf("4. Cancel a Booking\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                displaySeats(seats);
                break;
            case 2:
                displayAvailableSeats(seats);
                break;
            case 3:
                bookSeat(seats);
                break;
            case 4:
                cancelBooking(seats);
                break;
            case 5:
                printf("Exiting\n");
                break;
            default:
                printf("Invalid choice.\n");
        }
    }

    return 0;
}
```