FLIGHT RESERVATION SYSTEM (FRS)

*by Sabarigirish Manikandan (2410110283), and Ishaan Prakash (2410110429)*

The flight reservation system is a command line program that can create, modify, list, and (locally) store information about flights and flight bookings. FRS has a flight mode and a booking mode, where you can perform operations on or relating to flights and bookings.

Flights can be created with the following parameters: -

* Availability: whether the flight is accepting bookings
* Name: a name for the flight on or under 50 characters in length
* Seats: number of total seats in the flight
* Free seats: number of seats available for booking (is equal to total seats on creation)
* Source: 3 character code of the airport from which the flight is departing from
* Destination: 3 character code of the airport at which the flight will land after its journey

All parameters can be freely modified, except for free seats, which will be automatically updated upon booking creation and deletion. Flight availability is also automatically updated upon filling of seats.

Bookings can be created with the following parameters: -

* Passenger name: a name for the person under whom the flight ticket is booked (50 characters or under)
* Seats: number of seats they’d like to have booked under their name
* Age: age of the passenger under whom the ticket is booked
* Phone number: contact to be provided in case of intimation

All parameters can be freely modified, except for the number of seats booked. In order to change the number of seats being booked, a new booking must be created, deleting the original booking accordingly.

Flight and booking details are stored using file handling. These are read into an array of structs accordingly by functions that are called in the beginning of the program’s execution. These functions use pointers and dynamic memory allocation under the hood to ensure optimal memory usage. The program itself is an infinite loop that breaks when the user chooses to quit when prompted to, or if the program encounters any errors.

To avoid complication of the program’s structure, the functions have been placed in separate C source files with header files to be imported. This allows other programmers to use this code as a library. We also use the common JSON format to store our data, allowing for easy interoperability. This is done via the [cJSON](https://github.com/DaveGamble/cJSON) library written by David Gamble.