

BLG 102E - INTRODUCTION TO SCIENTIFIC AND ENGINEERING COMPUTATION

Assignment 2

Spring 2021

Objective: Making airline reservations for individuals and groups.

Description

In this assignment, you will be developing an airline reservation system that may be adapted to different flights operated by a variety of aircraft types with distinct layouts. The reservation system will allow individual and group reservations. The features and working principles of your program should follow the below-listed specifications.

1. Your program should first determine the seating layout of the specific aircraft that you are going to make reservations for. To this end, the number of seating rows (min: 1, max: 99), the number seats per row (min: 1, max: 20), the number of aisles (min: 1, max: 5), and the location of aisles are needed. The location of each aisle will be requested separately, and an aisle location will be specified as the number of seats to its left (here, left refers to the left of your screen). Once the layout information is provided by the user, your program will draw the layout on the screen as shown below. Each row is numbered numerically, and each column is marked with capital case letters in alphabetical order. Each dash ("-") represent an empty seat, and vertical bars ("|") mark the aisle borders.

```
Number of rows: 12
Number of seats per row: 7
Number of aisles: 2
Add aisle 1 after seat: 2
Add aisle 2 after seat: 5
```

	A	B				C	D	E				F	G
1	-	-				-	-	-				-	-
2	-	-				-	-	-				-	-
3	-	-				-	-	-				-	-
4	-	-				-	-	-				-	-
5	-	-				-	-	-				-	-
6	-	-				-	-	-				-	-
7	-	-				-	-	-				-	-
8	-	-				-	-	-				-	-
9	-	-				-	-	-				-	-
10	-	-				-	-	-				-	-

```
The number of people in the reservation
(0: print current reservations, -1: exit):
```

2. Below the layout, the program will ask the number of people for whom you want to make reservation. The program will not allow the customers to choose their seats. It will

automatically assign the seats. To determine the seat assignments, it will follow the below-listed rules.

- a. Considering the COVID-19 situation, your program will try to assign seats to the passengers to place them as far as possible from each other to minimize the risk of infection. To this end, for a passenger, it will assign a new seat that has the largest total distance to the already reserved seats. The distance between a seat s_1 (located at row i_1 , column j_1) and another seat s_2 (located at row i_2 , column j_2) will be computed as follows:

$$dist(s_1, s_2) = |i_1 - i_2| + \frac{|j_1 - j_2|}{2.0}$$

where each aisle location should also be taken into consideration as equivalent to one seat space.

$$adjusted_dist(s_1, s_2) = \begin{cases} 0 & \text{if } dist(s_1, s_2) = 0 \\ 100 - (100)^{\frac{1}{dist(s_1, s_2)}} & \text{otherwise} \end{cases}$$

Then, the score of a seat s_x is computed as:

$$score(s_x) = \sum_{s_i \in \text{Already reserved seats}} adjusted_dist(s_x, s_i)$$

- b. For a single passenger reservation, an empty seat with the maximum score is assigned. If there are different seat candidates with the same score, then the one with lower row number will be chosen. If the row numbers are also the same, the one with the lower column number will be assigned.
- c. For a family reservation that includes n individuals ($n > 1$), your program will look for a consecutive block of n seats located in the same row (note that a seat block is allowed to span one or more aisles). If there is no such block of seats, your program will not make group reservation by printing an appropriate message as illustrated in the following examples. For a block of n seats the distance-based score will be computed as follows:

$$seat - block - score(s_1, s_2, \dots, s_n) = \sum_{i=1}^n score(s_i)$$

If there are different seat blocks with the same score, then the one with the least number of aisles passing through the block will be assigned. For instance, seat blocks which are not divided with aisles in the same row will be preferred over those where one or more aisles divide the block. For two different seat blocks, if seat block scores are the same, and the number of aisles passing through these blocks are also the same, then the seat block with the lower row number will be preferred. If the row numbers are also the same, then the one with the lower column number (of the leftmost seat in the block) will be assigned.

- d. After an appropriate empty seat is assigned based on the above-explained score considerations, the row number and column letter of the assigned seat (s) will be listed

on the same line. Then, this message will be followed by the aircraft layout showing the just reserved seats as shown in the following example where X represent seats that are reserved in previous iterations, + shows the seats that are just reserved in the most recent step, and – shows empty seats.

```
The number of people in the reservation
(0: print current reservations, -1: exit): 5
Reserved seats: 8A 8B 8C 8D 8E
```

	A	B			C	D	E			F	G
1	X	-			-	-	-			-	X
2	-	-			-	-	-			-	-
3	-	-			X	X	X			X	-
4	-	X			-	-	-			-	-
5	-	-			-	-	-			X	-
6	-	-			-	-	-			-	-
7	-	-			X	-	-			-	-
8	+	+			+	+	+			-	-
9	-	-			-	-	-			-	X
10	X	-			-	-	-			-	-
11	-	-			-	-	-			-	-
12	X	X			X	-	-			-	X

```
The number of people in the reservation
(0: print current reservations, -1: exit):
```

3. If the requested number of reservations cannot be made, the program should notify the user as shown below.

	A	B			C	D	E			F	G
1	X	-			-	-	-			-	X
2	-	-			-	-	-			-	-
3	-	-			X	X	X			X	-
4	-	X			-	-	-			-	-
5	-	-			-	-	-			X	-
6	-	-			-	-	-			-	-
7	-	-			X	-	-			-	-
8	+	+			+	+	+			-	-
9	-	-			-	-	-			-	X
10	X	-			-	-	-			-	-
11	-	-			-	-	-			-	-
12	X	X			X	-	-			-	X

```
The number of people in the reservation
(0: print current reservations, -1: exit): 8
```

No available seats for the requested reservation!

```
The number of people in the reservation
(0: print current reservations, -1: exit):
```

4. After each reservation, the program will continue asking for information for another reservation. If the user enters 0 for the number of people in the reservation, the program will print the aircraft layout with the current reserved and empty seats properly marked. Please note that + signs will disappear and be replaced by X in the iterations that follow the current reservation iteration. When user enters -1, then the program will exit.
5. You may assume that we will always provide appropriate input, e.g., no string input where integer expected, or no values that are outside of the stated min-max boundaries in item 1.

Rules

- Your source code must have the name assignment2.c .
- Your program will be compiled using the following command on a Linux system. If it cannot be compiled and linked using this command, it will not be graded (failed submission).

```
gcc -std=c99 -Wall -Werror -lm assignment2.c -o assignment2
```
- Your program will be checked using an automatic checker. Therefore, make sure you print the messages exactly as given in the example runs. You will be given a Calico test for some basic input/output tests. Please run your assignment through Calico before submitting.
- Do NOT use statements for clearing the terminal or waiting for a keypress before exiting the program; these might cause your program to fail in the automatic tests. Some IDEs generate such statements, remove them. Running your program through Calico is the safest way to make sure that your program works as expected.
- Do NOT use any construct that hasn't been covered in the course before this week, such as structs.
- Do NOT use any external functions except for printf, scanf, abs, and pow.
- Make sure your coding style is proper and consistent. Use the clang-format tool if necessary. Don't use any variable names in a language other than English.
- This is an individual assignment. Collaboration in any form is NOT allowed. No working together, no sharing code in any form including showing code to your classmates or talking to them to give them ideas.
- All the code you submit must be your own. Don't copy/paste any piece of code from any resource including anything you've found on the Internet.
- The assignments will be checked for plagiarism using both automated tools and manual inspection. Any assignment involving plagiarism and/or infringement of intellectual property will be not be graded and is subject to further disciplinary actions.

Additional Example Runs:

Example Run 1:

```
Number of rows: 5
Number of seats per row: 4
Number of aisles: 1
Add aisle 1 after seat: 2
```

	A	B				C	D
1	-	-				-	-
2	-	-				-	-
3	-	-				-	-
4	-	-				-	-
5	-	-				-	-

```
The number of people in the reservation
(0: print current reservations, -1: exit): 1
Reserved seats: 1A
```

	A	B				C	D
1	+	-				-	-
2	-	-				-	-
3	-	-				-	-
4	-	-				-	-
5	-	-				-	-

```
The number of people in the reservation
(0: print current reservations, -1: exit): 1
Reserved seats: 5D
```

	A	B				C	D
1	X	-				-	-
2	-	-				-	-
3	-	-				-	-
4	-	-				-	-
5	-	-				-	+

```
The number of people in the reservation
(0: print current reservations, -1: exit): 1
Reserved seats: 2D
```

	A	B				C	D
1	X	-				-	-
2	-	-				-	+
3	-	-				-	-
4	-	-				-	-
5	-	-				-	X

```
The number of people in the reservation
(0: print current reservations, -1: exit): 1
Reserved seats: 4A
```

	A	B			C	D
1	X	-			-	-
2	-	-			-	X
3	-	-			-	-
4	+	-			-	-
5	-	-			-	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3B

	A	B			C	D
1	X	-			-	-
2	-	-			-	X
3	-	+			-	-
4	X	-			-	-
5	-	-			-	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 1C

	A	B			C	D
1	X	-			+	-
2	-	-			-	X
3	-	X			-	-
4	X	-			-	-
5	-	-			-	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 2
 Reserved seats: 5A 5B

	A	B			C	D
1	X	-			X	-
2	-	-			-	X
3	-	X			-	-
4	X	-			-	-
5	+	+			-	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 3
 Reserved seats: 2A 2B 2C

	A	B			C	D
1	X	-			X	-
2	+	+			+	X
3	-	X			-	-
4	X	-			-	-
5	X	X			-	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 2

Reserved seats: 4C 4D

	A	B				C	D
1	X	-				X	-
2	X	X				X	X
3	-	X				-	-
4	X	-				+	+
5	X	X				-	X

The number of people in the reservation

(0: print current reservations, -1: exit): 3

No available seats for the requested reservation!

The number of people in the reservation

(0: print current reservations, -1: exit): 1

Reserved seats: 3D

	A	B				C	D
1	X	-				X	-
2	X	X				X	X
3	-	X				-	+
4	X	-				X	X
5	X	X				-	X

The number of people in the reservation

(0: print current reservations, -1: exit): 2

No available seats for the requested reservation!

The number of people in the reservation

(0: print current reservations, -1: exit): 1

Reserved seats: 1D

	A	B				C	D
1	X	-				X	+
2	X	X				X	X
3	-	X				-	X
4	X	-				X	X
5	X	X				-	X

The number of people in the reservation

(0: print current reservations, -1: exit): 1

Reserved seats: 5C

	A	B				C	D
1	X	-				X	X
2	X	X				X	X
3	-	X				-	X
4	X	-				X	X
5	X	X				+	X

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
Reserved seats: 3A
```

	A	B			C	D
1	X	-			X	X
2	X	X			X	X
3	+	X			-	X
4	X	-			X	X
5	X	X			X	X

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
Reserved seats: 1B
```

	A	B			C	D
1	X	+			X	X
2	X	X			X	X
3	X	X			-	X
4	X	-			X	X
5	X	X			X	X

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
Reserved seats: 4B
```

	A	B			C	D
1	X	X			X	X
2	X	X			X	X
3	X	X			-	X
4	X	+			X	X
5	X	X			X	X

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
Reserved seats: 3C
```

	A	B			C	D
1	X	X			X	X
2	X	X			X	X
3	X	X			+	X
4	X	X			X	X
5	X	X			X	X

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
```

No available seats for the requested reservation!

The number of people in the reservation

```
(0: print current reservations, -1: exit): 1
```

No available seats for the requested reservation!


```
The number of people in the reservation
(0: print current reservations, -1: exit):
```

Example Run 2:

```
Number of rows: 4
Number of seats per row: 5
Number of aisles: 2
Add aisle 1 after seat: 1
Add aisle 2 after seat: 4
```

	A				B	C	D				E
1	-				-	-	-				-
2	-				-	-	-				-
3	-				-	-	-				-
4	-				-	-	-				-

```
The number of people in the reservation
(0: print current reservations, -1: exit): 1
Reserved seats: 1A
```

	A				B	C	D				E
1	+				-	-	-				-
2	-				-	-	-				-
3	-				-	-	-				-
4	-				-	-	-				-

```
The number of people in the reservation
(0: print current reservations, -1: exit): 4
Reserved seats: 4B 4C 4D 4E
```

	A				B	C	D				E
1	X				-	-	-				-
2	-				-	-	-				-
3	-				-	-	-				-
4	-				+	+	+				+

```
The number of people in the reservation
(0: print current reservations, -1: exit): 5
Reserved seats: 2A 2B 2C 2D 2E
```

	A				B	C	D				E
1	X				-	-	-				-
2	+				+	+	+				+
3	-				-	-	-				-
4	-				X	X	X				X

```
The number of people in the reservation
(0: print current reservations, -1: exit): 6
```

No available seats for the requested reservation!

The number of people in the reservation
 (0: print current reservations, -1: exit): 2
 Reserved seats: 1D 1E

	A	B	C	D	E
1	X	-	-	+	+
2	X	X	X	X	X
3	-	-	-	-	-
4	-	X	X	X	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3A

	A	B	C	D	E
1	X	-	-	X	X
2	X	X	X	X	X
3	+	-	-	-	-
4	-	X	X	X	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3E

	A	B	C	D	E
1	X	-	-	X	X
2	X	X	X	X	X
3	X	-	-	-	+
4	-	X	X	X	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 4A

	A	B	C	D	E
1	X	-	-	X	X
2	X	X	X	X	X
3	X	-	-	-	X
4	+	X	X	X	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3C

	A	B	C	D	E
1	X	-	-	X	X
2	X	X	X	X	X
3	X	-	+	-	X
4	X	X	X	X	X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 1B

	A		B	C	D		E
1	X		+	-	X		X
2	X		X	X	X		X
3	X		-	X	-		X
4	X		X	X	X		X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3B

	A		B	C	D		E
1	X		X	-	X		X
2	X		X	X	X		X
3	X		+	X	-		X
4	X		X	X	X		X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 3D

	A		B	C	D		E
1	X		X	-	X		X
2	X		X	X	X		X
3	X		X	X	+		X
4	X		X	X	X		X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1
 Reserved seats: 1C

	A		B	C	D		E
1	X		X	+	X		X
2	X		X	X	X		X
3	X		X	X	X		X
4	X		X	X	X		X

The number of people in the reservation
 (0: print current reservations, -1: exit): 1

No available seats for the requested reservation!

The number of people in the reservation
 (0: print current reservations, -1: exit):