

Modified June 22, 2015.

The [Miami Dade College Wolfson Latin Jazz Big Band](#) needs a program to organize where the musicians will stand when they play at away games. Each away stadium is different, so when they arrive the [conductor](#) gets the following information from the local organizer:

The number of rows they have to stand on. The maximum number of rows is 10. The rows are labelled with capital letters, 'A', 'B', 'C', etc.

For each row, the number of positions in the row. The maximum number of positions is 8. The positions are numbered with integers, 1, 2, 3, etc.

The conductor then starts assigning people to positions, but is constrained by weight limits: Musicians, fully clothed and holding their instruments, weigh from [45kg to 200kg](#), and the total weight of a row may not exceed 100kg per position (e.g., a row with 5 positions may not have more than 500kg of musicians on it). The conductor wants a program that allows musicians to be added and removed from positions, while ensuring the constraints are all met. At any stage the conductor wants to be able to see the current assignment - the weight in each position (0kg for vacant positions) and the total & average weight for each row.

The program must be menu driven, with options to:

Add a musician (by weight) to a vacant position.

Remove a musician from an occupied position.

Print the current assignment.

Exit (so the musicians can [start playing](#))

The program must be reasonably idiot proof:

Menu options must be accepted in upper and lower case.

Row letters must be accepted in upper and lower case.

All input must be checked to be in range, and if not the user must be asked to input again.

You may assume that numeric input will be syntactically correct.

Here's what a sample run should look like (with the keyboard input shown in *italics*) ...

```
Welcome to the Latin Jazz Big Band
-----
Please enter number of rows                : 11
ERROR: Out of range, try again            : 3
Please enter number of positions in row A : -4
ERROR: Out of range, try again            : 3
Please enter number of positions in row B : 4
Please enter number of positions in row C : 5

(A)dd, (R)emove, (P)rint,                e(X)it : p

A:  0.0  0.0  0.0                        [  0.0,    0.0]
B:  0.0  0.0  0.0  0.0                    [  0.0,    0.0]
C:  0.0  0.0  0.0  0.0  0.0              [  0.0,    0.0]

(A)dd, (R)emove, (P)rint,                e(X)it : a
```

```

Please enter row letter           : Q
ERROR: Out of range, try again   : B
Please enter position number (1 to 4) : 6
ERROR: Out of range, try again   : 1
Please enter weight (45.0 to 200.0) : 22
ERROR: Out of range, try again   : 222
ERROR: Out of range, try again   : 180
***** Musician added.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : p
```

```

A:   0.0   0.0   0.0                [   0.0,   0.0]
B: 180.0   0.0   0.0   0.0          [ 180.0,  45.0]
C:   0.0   0.0   0.0   0.0   0.0    [   0.0,   0.0]

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : A
```

```

Please enter row letter           : b
Please enter position number (1 to 4) : 1
ERROR: There is already a musician there.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : A
```

```

Please enter row letter           : B
Please enter position number (1 to 4) : 2
Please enter weight (45.0 to 200.0) : 150
***** Musician added.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : p
```

```

A:   0.0   0.0   0.0                [   0.0,   0.0]
B: 180.0 150.0   0.0   0.0          [ 330.0,  82.5]
C:   0.0   0.0   0.0   0.0   0.0    [   0.0,   0.0]

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : a
```

```

Please enter row letter           : b
Please enter position number (1 to 4) : 3
Please enter weight (45.0 to 200.0) : 175
ERROR: That would exceed the average weight limit.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : a
```

```

Please enter row letter           : b
Please enter position number (1 to 4) : 4
Please enter weight (45.0 to 200.0) : 55
***** Musician added.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : p
```

```

A:   0.0   0.0   0.0                [   0.0,   0.0]
B: 180.0 150.0   0.0  55.0          [ 385.0,  96.3]
C:   0.0   0.0   0.0   0.0   0.0    [   0.0,   0.0]

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : r
```

```

Please enter row letter           : b
Please enter position number (1 to 4) : 3
ERROR: That position is vacant.

```

```
(A)dd, (R)emove, (P)rint,          e(X)it : r
```

```

Please enter row letter           : b
Please enter position number (1 to 4) : 1
***** Musician removed.

```

```

(A)dd, (R)emove, (P)rint,          e(X)it : p
A:   0.0   0.0   0.0                [   0.0,   0.0]
B:   0.0 150.0   0.0  55.0          [ 205.0,  51.3]
C:   0.0   0.0   0.0   0.0   0.0    [   0.0,   0.0]

(A)dd, (R)emove, (P)rint,          e(X)it : q
ERROR: Invalid option, try again    : P

A:   0.0   0.0   0.0                [   0.0,   0.0]
B:   0.0 150.0   0.0  55.0          [ 205.0,  51.3]
C:   0.0   0.0   0.0   0.0   0.0    [   0.0,   0.0]

(A)dd, (R)emove, (P)rint,          e(X)it : X

```

You must ...

Do a top down design for this problem, producing a structure chart (it's OK to draw it by hand).

Do a design for the program.

Implement the program in Java (you must use a two dimensional array to store the weight assignments).

The assignment is due at the start of your lab session in the week starting 29th June. You must submit:

- Structure chart
- Design
- Printout of code

Your project is worth 15% of the course grade. It will be assessed on:

- Structure chart - 1.5%
- Design - 1.5%
- Performance to specification - 10%
- Programming style - 2%

Last modified June 22, 2015