

Matrix Questions

Part (a)

Write a function *DrawBorder*, as stated below. *DrawBorder* should set the following elements of the matrix *Screen* to the given character *c*:

1. All of the elements in row *j*, from column *j* to column *Screen.numcols() - j - 1*.
2. All of the elements in row *Screen.numrows() - j - 1*, from column *j* to column *Screen.numcols() - j - 1*.
3. All of the elements in column *j*, from row *j* to row *Screen.numrows() - j - 1*.
4. All of the elements in column *Screen.numcols() - j - 1*, from row *j* to row *Screen.numrows() - j - 1*.

Assume that, as specified in *DrawBorder*'s precondition, both *Screen.numrows()* and *Screen.numcols()* are greater than $j * 2$.

For example, assume that *Screen* is a 6-by-7 matrix, containing only dots. Here are some examples that illustrate calls to *DrawBorder*:

| Function Call | Resulting Value of <i>Screen</i> |
|----------------------------|--|
| DrawBorder(Screen, 0, 'X') | <pre>X X X X X X X X X X X X X X X X X X X X X X</pre> |
| DrawBorder(Screen, 1, 'X') | <pre>. X X X X X . . X . . . X . . X . . . X . . X X X X X</pre> |
| DrawBorder(Screen, 2, 'X') | <pre>. X X X X X X</pre> |

Complete function *DrawBorder* below. Assume that it is called only with values that satisfy its precondition.

```
void DrawBorder(apmatrix<char> & Screen, int j, char c)
// pre: Screen.numrows() > j * 2
//      Screen.numcols() > j * 2
```

Part (b)

Write function *OutermostBorder*, as stated below. Function *OutermostBorder* should find and return the position of the upper-left corner of the outermost border in matrix *Screen*; that is, the smallest number j such that $Screen.numrows() > j * 2$ and $Screen.numcols() > j * 2$ and all of the following elements of matrix *Screen* contain the same character.

1. All of the elements in row j , from column j to column $Screen.numcols() - j - 1$.
2. All of the elements in row $Screen.numrows() - j - 1$, from column j to column $Screen.numcols() - j - 1$.
3. All of the elements in column j , from row j to row $Screen.numrows() - j - 1$.
4. All of the elements in column $Screen.numcols() - j - 1$, from row j to row $Screen.numrows() - j - 1$.

If there are no borders in *Screen*, *OutermostBorder* should return -1.

For example:

| <u>Value of <i>Screen</i></u> | <u>Result of the call <i>OutermostBorder</i>(<i>Screen</i>)</u> |
|--|---|
| X X X X X X X X X X X X X X X X X X X X X X | 0 |
| a X X X X X X X o o o o o X X o . . . o X X o . . . o X X o o o o o X X X X X X X X | 1 |
| a X X X X X X X b o o o o X X o . . . o X X o . . . o X X o o o o o X X X X X X X X | 2 |
| a X X X X X X X b o o o o X X o c . . o X X o . . . o X X o o o o o X X X X X X X X | -1 |

Complete function *OutermostBorder* below

```
int OutermostBorder(const apmatrix<char> & Screen)
```