CSc 110, Autumn 2016

Lecture 24: Lists of Lists

Mountain peak

Write a program that reads elevation data from a file, draws it on a `DrawingPanel` and finds the path from the highest elevation to the edge of the region.

Data:

```
34 76 87 9 34 8  22 33 33 33 45 65 43 22
5  7  88 0 56 76 76 77 4  45 55 55 4  5
…
```

This data is a different shape. How should we store it?

Lists of lists

- You can put a list in a list

```
list = [[1, 2, 3], [4, 5, 6]]
```

How can you access 2?

```
list[0][1]
```

How can you find the length of the second inner list (`[4, 5, 6]`)?

```
len(list[1])
```

List of lists mystery

```
def mystery(data, pos, n):
    result = []
    for i in range(0, n):
        for j in range(0, n):
            result.append(data[i + pos][j + pos])
    return result
```

Suppose that a variable called grid has been declared as follows:

```
grid = [[8, 2, 7, 8, 2, 1], [1, 5, 1, 7, 4, 7],
        [5, 9, 6, 7, 3, 2], [7, 8, 7, 7, 7, 9],
        [4, 2, 6, 9, 2, 3], [2, 2, 8, 1, 1, 3]]
```

which means it will store the following 6-by-6 grid of values:

8	2	7	8	2	1
1	5	1	7	4	7
5	9	6	7	3	2
7	8	7	7	7	9
4	2	6	9	2	3
2	2	8	1	1	3

Function Call	Contents of List Returned
`mystery(grid, 2, 2)`	_____
`mystery(grid, 0, 2)`	_____
`mystery(grid, 3, 3)`	_____

For each call at right, indicate what value is returned. If the function call results in an error, write error instead.

Mountain peak

Write a program that reads elevation data from a file, draws it on a `DrawingPanel` and finds the path from the highest elevation to the edge of the region.

Data:

```
34 76 87 9 34 8  22 33 33 33 45 65 43 22
5  7  88 0 56 76 76 77 4  45 55 55 4  5
…
```

Creating Lists of lists

- `list = [[0] * 4] * 5` will NOT create a list of lists
  - This will create a list with 5 spots that all contain the SAME list that is 4 long.

- Instead, write the following:

```
list = []
for i in range(0, 5):
    list.append([0] * 4)
```

Days till

- Write a function called `days_till` that accepts a start month and day and a stop month and day and returns the number of days between them

call	return
`days_till("december", 1, "december", 10)`	9
`days_till("novembeR", 15, "december", 10)`	25
`days_till("OCTober", 6, "december", 17)`	72
`days_till("october", 6, "ocTober", 1)`	360

Days till solution

```
def days_till(start_month, start_day, stop_month, stop_day):
    months = (('january', 31),('february', 28),('march', 31),('april', 30), ('may', 31),('june', 30),
              ('july', 31), ('august', 31),('september', 30), ('october', 31), ('november', 30), ('december', 31))

    if start_month.lower() == stop_month.lower() and stop_day >= start_day:
        return stop_day - start_day
    days = 0
    for i in range(0, len(months)):
        month = months[i]
        if month[0] == start_month.lower():
            days = month[1] - start_day
            i += 1
            while months[i % 12][0] != stop_month.lower():
                days += months[i % 12][1]
                i += 1
            days += stop_day
    return days
```