Lists of lists! Create and print these two-dimensional structures.

1 Your assignment

Do the following:

1. Create a new file for yourself, lab7.py, in which to do your programming. Be sure that it ends with the boilerplate that calls the function main() in order to get your program started:

   ```
   if __name__ == '__main__':
       main()
   ```

   Of course, this step implies that your module will contain a main() function that calls on the other functions that you write.

2. Write a function print_grid(g) that accepts a list of lists as a parameter, and then prints all of those items in a grid format. For example, if we do the following from (say) main()...

   ```
   def main():
       l = [ [11, 12, 13, 14], [21, 22, 23, 24], [31, 32, 33, 34] ]
       print_grid(l)
   ```

   ...then the print_grid function should display the following:

   ```
   11 12 13 14
   21 22 23 24
   31 32 33 34
   ```

3. Write a function make_x_table(rows, cols) that creates a grid of numbers that is rows tall and cols wide, such that each element at a given row r and a given column c is the value v = r*c. It must create this table of numbers as a list of lists, and then return that structure; it should not print anything.

   For example, if we call make_x_table(3, 5), then we expect a list of lists to be returned with the following values and structure:
4. Write a function `make_pascal(rows)` that creates a 2-D list structural that stores the leading rows of Pascal's triangle. For example, for `rows = 5`, this function should make the following list of lists:

```
[ [ 1 ],
  [ 1, 1 ],
  [ 1, 2, 1 ],
  [ 1, 3, 3, 1 ],
  [ 1, 4, 6, 4, 1 ] ]
```

That is, the number on row \( r \) and column \( c \) is the sum of the values on row \( r - 1 \) at columns \( c - 1 \) and \( c \). The edge numbers are always 1.

5. EXTRA CHALLENGE: Modify your function `print_grid(g)` such that it makes each printed value in its grid of the same width. That is, assume all of the values to be printed are integers; find the widest integer to print, and then make sure, when printing, that every entry occupies that number of spaces. Thus, in spite of different width values, the grid is nicely aligned.

With all of these functions written, write `main()` such that it will demonstrate that each function does its job properly. You should use your `print_grid` function to show the results returned by `make_x_table` and `make_pascal`.

2 Submitting your work

Go to the CS submission system to submit your work for this lab. You need only submit your `lab7.py` module.

This assignment is due on Wednesday, Mar-26, 11:59 pm