**Student Name:** **Student ID:**

# Important Information

* For *every* lab and home assignment, store all your work in your personal repository in a subdirectory named **mXX**, where XX is the module number. Carefully name the program as described in each problem.
* Your programs are extracted from your repository by a Python script. If there are any errors in the program name, then your instructor will never see your program, and you will receive a mark of zero.
* Push your work to the server often, and ensure that you push the final version of a program by the deadline specified, because the script extracting them can be run at any time after the deadline.

# Instructions

1. Read chapters 1–5 in *How to Think Like a Computer Scientist: Learning with Python*, available at [www.greenteapress.com/thinkpython/thinkCSpy.pdf](http://www.greenteapress.com/thinkpython/thinkCSpy.pdf).
2. Complete Problems 9 and 10.

**Note:** Problem numbering continues from the module’s lab.

## Problem 9

Write a Python program named **m02p09.py** that generates a sine wave on the terminal using the ASCII star character.

* Make the amplitude of the sine wave 10 lines, and make one period of the sine wave 60 points (60 characters).
* Draw the zero axis using the dash character.
* Create an empty list of lines, fill each line with 60 spaces, and then place the stars and dashes.

The output should look like this:

\*\*\*\*\*\*\*

\*\* \*\*

\* \*

\*\* \*\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*-----------------------------\*-----------------------------

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*\* \*\*

\* \*

\*\* \*\*

\*\*\*\*\*\*\*

## 

lines=[]

lines\_rvs=[]

space=' '

star='\*'

dash='-'

length=30

lines.append((space\*12)+(star\*7))

fsp=10 #number of spaces in front of stars

msp=7 #number of spaces between stars

for i in range (2):

lines.append(space\*fsp+star\*2+space\*msp+star\*2)

fsp-=1 #next line 1 star move forward

msp+=4 #last line 2 star, make 2 space on each side 2\*2

lines.append(space\*fsp+star\*1+space\*msp+star\*1)

fsp-=2 #next line 2 stars move forward

msp+=2 #last line 1 star, make 1 space on each side 1\*2

fsp+=1 #last loop moved 2 space forward, but next line only move 1 star forward

for i in range (5):

lines.append(space\*fsp+star\*1+space\*msp+star\*1)

fsp-=1

msp+=2

lines.append((star+dash\*29)\*2) # the zero axis.

for line in lines:

print line

for i in range(9,-1,-1): #print backwords of list,without the zero axis, which is the last element

print space\*length+lines[i] #print 30 spaces before lines

"""lines\_rvs = lines[:] #hard copy list, pop last str, then reverse list.

lines\_rvs.pop()

lines\_rvs.reverse()

for line in lines\_rvs:

print space\*length+line"""

## Problem 10

Write a Python program named **m02p10.py** that prints rows of Pascal’s triangle numbers in the shape of a triangle.

* Make the printing stop when the row would exceed 80 characters.
* Define the function pascal(row), which takes as a parameter the list of numbers in a row, and returns a list of numbers for the next row.

The output should look like this:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1…

linelen=80

def pascal(row): #calculate next row

nextrow=[1] #first element of new row is 1

for i in range(1,(len(row))): #contains one more element than previous row

nextrow.append(row[i-1]+row[i]) #generate elements in middle

nextrow.append(1)

return nextrow

onerow=[1]

while 1:

rowstr=''

for num in onerow:

rowstr += str(num) + ' '

if len (rowstr) >= linelen:

break

print (" "\*((linelen-len(rowstr))/2)), #print spaces in front of row

print rowstr

onerow=pascal(onerow) #get nextrow