**Student Name/Grade: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Rubric:**

\_\_\_\_\_\_\_/20 Tells a story and reads user input to branch out

\_\_\_\_\_\_\_/15 At least 2 branches with at least 4 choices each, at least one branch with at least 3 options

\_\_\_\_\_\_\_/5 Meaningful variable names, comments, overall style

\_\_\_\_\_\_\_/10 File named properly and turned in on time

\_\_\_\_\_\_\_/5**EC** Extra Credit: if/else can handle multiple versions of input

\_\_\_\_\_\_\_/50 Total

**Description:**

Remember Choose Your Own Adventure books? Here's your chance to write your own. Write a text-based adventure that gives the user options for how the story plays out. Output a little bit of story (use new lines frequently so you don’t have to scroll around forever) and then ask the user for what they want to do next, and make it obvious what the choices are. For instance, my story might say “You come across a troll at a bridge. Do you -talk- or -fight-?” That makes it obvious that the user should either type in “talk” or “fight.” Read their input, then use if/else statements to send the story down different paths. If they don’t type exactly the right thing, it’s OK if the program ends right away. Don’t worry about checking until they do it right. The story can be about absolutely anything you want (as long as it's school appropriate). Just make sure there are at least 2 paths that are at least 4 choices deep (it's fine if some branches end happily ever after or result in death/an ending right away), and have at least one path with at least 3 options (use an elif). Because it's going to be tough to keep track of everything you also need to have a paper copy of a flowchart showing how the branches fit together, and that’s worth a **homework grade on Tuesday 8/29**.

**Extra Credit:**

If at least one of your choices can handle multiple versions of inputs (for example, “Left” and “left” both work), I’ll give you five extra credit points

**Optional hint:** As a way of making your code less crazily indented, you might consider writing each chunk of text as its own function. For example, you could write something like this:

def bridge():

print some stuff explaining the story

ask them for their choice

if they chose right:

bridge\_right()

if they chose left:

bridge\_left()

def bridge\_right():

more story

ask for choice

if it's talk:

talk()

if it's fight:

fight()

def bridge\_left():

stuff here

def talk():

stuff here

def fight():

stuff here

etc.

It just keeps it from looking like this:

Bridge:

If they go right:

If they fight:

If they make another choice:

At this point you’re so indented it’s hard to read

**Program Requirements**

* Must be written in Python (all or nothing)
* Must compile without modifications to the code (all or nothing)
* The file is named **adventure\_game\_lastname.py** and turned in on time (10 points)
* At the top of your code, you must include a block comment:
  + Your Name
  + Date
  + Class/Instructor
  + Project number and filename
  + One-line description of the purpose of the file

**What to do with it once you’re done (10 points)**

* Email me a copy of your .py file
  + The subject line MUST be “[ICS] Week 04 LastName” so mine would be “ [ICS] Week 04 Swindle”
  + MUST be turned in by the start of class on 9/5 or it’s late.