AP Computer Science Principles: 2020 Create Task
Student Sample H

3a.

This video illustrates my program running in its entirety. The input in my program would be the user input when prompted with questions. The output would be the program telling if the water is safe for drinking or for animals to live in. The purpose of my program is to help people identify if their water is safe for drinking or for animals. People can enter information about their water into the program and it will let them know if they should drink it.

3b.

```
values []
```

```
nitloop = True
while nitloop == True:
  try:
    nitrates = int(raw_input("How many parts per liter of
    nitrates are there?"))
    nitloop = False
    values.append(nitrates)
  except ValueError:
    print "Please enter an integer!"
```

In this example above, the list “values” is not complete yet. There are still parts of the code adding more bits to the list. The list is being added onto with the .append function. This manages complexity because it allows just one variable to store multiple integers and float values. Without the list, my code would be much longer and I would have to write out each variable every time I wanted to use it.

```
#this will be for nitrates
if values[0] == 10:
  print "The nitrate levels are %s for fish and to drink"
  %safe
elif values[0] >= 5:
  if values[0] < 10:
    print "The nitrate levels are %s for fish only" %safe
else:
  print "The nitrate levels are %s for fish or to drink"
  %notsafe
```

In this example, my list is being evaluated. This is able to manage complexity because I have to rewrite it multiple times. Instead of having to write every variable, I can just write the index of the list. This is very useful because I use it in many different places in my code. Without this list, my code would be much messier and it would be more difficult to read and debug.

3c.

```
nitloop = True
while nitloop == True:
  try:
    nitrates = int(raw_input("How many parts per liter of
    nitrates are there?"))
    nitloop = False
    values.append(nitrates)
  except ValueError:
    print "Please enter an integer!"
```

This algorithm plays a vital role in my program. This is because it takes in the user input and adds it to the list. Without this code, the entire program would not be able to run. This is because this makes up what goes into the list and what the list evaluates. When missing, the code is not able to evaluate all the sources correctly. This algorithm accomplishes the task of adding user input into a list. It does this by asking the user to enter how many parts per liter of nitrates there are, and then adding it to a list. If the user enters something other than an integer, it will ask the user to try again.

3d.

Two different test cases are if you entered “6” or “10.” When 6 is entered into the program, it will take that number and add it to a list of values. Once that is done, the second half of the code will test to see if the number is equal to 10. Since 6 is not equal to 10 it moves on. Then it tests to see if the number is greater or equal to 5 and less than 10. Since it is true, the program will print “The nitrate levels are safe for fish only.” When 10 is entered in the program, it will once again take this number and add it to the list of values. After that, it will test to see if that number is equal to 10. Because 10 is equal to 10. It will print out “The nitrate levels are safe for fish and to drink.”