```python
#!/usr/bin/env python
# coding: utf-8

# In[23]:


"""
Instruction:
- Read this script thoroughly to understand its structure before you attempt to do anything.
- Do not modify anything in this script unless instructed otherwise.
- Replace <your full name> below with your full name.
- Your solution to each problem goes between the line marked as "Student solution code . . . starts here"
   and a return statement. The parameter to each return statement needs to be replaced with the correct one.
- The indented lines start with four space characters. Do not change them. Your solution lines must be
   indented also in the same way.
- Frequently test/execute this script by issuing the "python3 hw1.py" command in a terminal.
- You can also grade your solution on your own by executing the grade_hw1.py script, i.e., "python3 grade_hw1.py".
   Do not change any part of grade_hw1.py lest it should result in wrong output or errors.
"""

"""
CSC 1611 Problem Solving with Python
Homework 1
Noah Amabile Foilb
"""

WRONG_ANSWER = -1

# Problem 1
def celsius_to_fahren(cel):
      """ Returns the quantity in Fahrenheit equivalent to cel given in Celsius. """
      """ Student solution code, which can be of multiple lines of instructions, starts here. """
      fahren = round(1.8*cel + 32,2)



      # Update the parameter of return with the correct expressions.
      return fahren

# Problem 2
def degree_to_radian(degree):
      """ Returns the quantity in radians equivalent to degree given in degrees. """
      """ Student solution code, which can be of multiple lines of instructions, starts here. """
      import math
      radian = round(degree*math.pi/180,10)

      # Update the parameter of return with the correct expressions.
      return radian

# Problem 3
def cartesian_distance(start_radian, target_radian, radius ):
      """ Returns the start Cartesian x,y coordinates, the target Cartesian x,y
      coordinates, and the distance between the start and the target coordinates
      in the given order, given the start_ and target_radian angular positions
      in degrees. """
      """ Student solution code, which can be of multiple lines of instructions, starts here. """
      import math
      start_x = round(radius*math.cos(start_radian),4)
      start_y = round(radius*math.sin(start_radian),4)
      target_x = round(radius*math.cos(target_radian),4)
      target_y = round(radius*math.sin(target_radian),4)
      distance = round(math.sqrt((target_x - start_x)**2+(target_y - start_y)**2),4)

      # Update the parameter of return with the correct expressions.
      return start_x,start_y,target_x,target_y,distance

"""
The following code is provided to help you understand how to test your solutions.
Feel free to assign different values to the variables used below to test your solution
with different input values.
"""
if __name__=='__main__':
      print("\nProblem 1")
```

```python
cel = -273.15 # Assign to cel the value which you want to test the function with.
fahren = celsius_to_fahren(cel)
print(f'{cel} degrees Celsius is equivalent to {fahren} degrees Fahrenheit.')

print("\nProblem 2")
deg = 360 # Assign to deg the value which you want to test the function with.
rad = degree_to_radian(deg)
print(f'{deg} degrees is equivalent to {rad} radians.')

print("\nProblem 3")
sa = -162  # Assign to sa and ta the values which you want to test the function with.
ta = -306
sx,sy,tx,ty,d = cartesian_distance(degree_to_radian(sa),degree_to_radian(ta),1)
print(f'start angle:{sa}, target angle:{-ta}, start coord: ({sx},{sy}), target coord: ({tx},{ty}), dist:{d}')
```