```
# EXAMPLE SOURCE CODE EVENT

# INSTRUCTIONS: Write your python code in the following functions to programmatically solve the
# problem given in each comments section. There may be several possible ways to obtain the solution to a
# problem.
# Multiple choice problems should return only the correct letter.
# NOTE: Actual event will typically include many more problems, and will have an increasing difficulty as
# demonstrated here. Problem 6 is an example of the highest difficulty you might encounter.

# PROBLEM 0:
# Points: 2
# Which python operator returns the remainder of a division equation.
# Return only the correct letter.
def problem0():
        # A) <>
        # B) /=
        # C) /
        # D) %

        return ""

# PROBLEM 1:
# Points: 2
# Which of the following statements will return False. x = 10, y = 20
# Return only the correct letter.
def problem1():
        # A) (x != y and not(x > y))
        # B) not(x == y) or ((y / 3) == x)
        # C) (x <= y and not((x + x) == y))
        # D) (x**2 == y and y == x) or (x*2 >= y)

        return ""


# PROBLEM 2:
# Points: 3
# Find the sum of the positive, even numbers between 0 and 100.
# Write your code and return the sum.
def problem2():

        return ""




# Problem 3:
# Points: 3
# Find the amount of prime numbers between 2 and 10000.
# Write your code and return the number.
def problem3():

        return ""




# Problem 4:
# Points: 3
# Bonus Objective: Use a while loop. Points: 1
# Using the given string, calculate the sum of the ord() value of each character. The ord() function returns
# the numerical value of a single character.
# Write your code and return the sum.
```

```
def problem4():
        str = "How much is this string of characters, '!#)!^$!', actually worth?"

        return ""



# Problem 5:
# Points: 4
# Find the smallest positive integer number that has 50 divisors.
# Write your code and return the number.
def problem5():

        return ""



# Problem 6:
# Points: 8
# Bonus Objective: Include comments within your code. Points: 1
# Consider a number n is backwards if it is written in reverse order.
# 34 written backwards is 43 and 103 written backwards is 301. Using positive
# integers, find the total number of backwards numbers less than one-million
# where the sum of the number forwards and backwards contains only
# odd digits (34+43=77).
# Write your code and return the answer.
def problem6():

        return ""



# Main Function. *** DO NOT EDIT ***
# This function calls each problem's function and prints the returned value. If this
# function is modified to print data differently, the team's program may not be scored.
# 27 Points total
def main():
   print("Problem 0: "+str(problem0()))
   print("Problem 1: "+str(problem1()))
   print("Problem 2: "+str(problem2()))
   print("Problem 3: "+str(problem3()))
   print("Problem 4: "+str(problem4()))
   print("Problem 5: "+str(problem5()))
   print("Problem 6: "+str(problem6()))
main()
```