Q1.

The following is a program to find the area of a triangle, given its sides as input. Copy-paste the code down, and make appropriate corrections so that the code works for the test cases. Note that the output should have only two digits in the fractional part.

\*\* DO NOT REWRITE THE ENTIRE CODE. ONLY UPDATE THE PLACES WHERE YOU FIND AN ERROR.

\*\* DO NOT INCLUDE NEW MODULES OR EXTRA FUNCTIONS

a,b,c=input().split()

a=int(a)

b=int(b)

c=int(c)

s = (a + b + c) // 2

area = (s\*(s-a)\*(s-b)\*(s-c)) \* 0.5

print('The area of the triangle is %0.5f' %area)

5 6 8

The area of the triangle is 14.98

7 8 9

The area of the triangle is 26.83

50 70 89

The area of the triangle is 1745.15

ANS:

a,b,c=input().split()

a=int(a)

b=int(b)

c=int(c)

s = (a + b + c) / 2

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

print('The area of the triangle is %0.2f' %area)

Q2.

The following is a program to display all the prime numbers within an interval. Copy-paste the code down, and make appropriate corrections so that the code works for the test cases.

\*\* DO NOT REWRITE THE ENTIRE CODE. ONLY UPDATE THE PLACES WHERE YOU FIND AN ERROR.

\*\* DO NOT INCLUDE NEW MODULES OR EXTRA FUNCTIONS

a,b=input().split()

lower=int(a)

upper=int(b)

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):

# all prime numbers are greater than 1

if num > 1:

for i in range(1, num):

if (num % i) != 0:

continue

else:

print(num)

1 40

Prime numbers between 1 and 40 are:

2

3

5

7

11

13

17

19

23

29

31

37

50 100

Prime numbers between 50 and 100 are:

53

59

61

67

71

73

79

83

89

97

Ans:

a,b=input().split()

lower=int(a)

upper=int(b)

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):

# all prime numbers are greater than 1

if num > 1:

for i in range(2, num):

if (num % i) == 0:

break

else:

print(num)

Q3. The following is a program to print the HCF(Highest Common Factor) of two numbers taken as input. Copy-paste the code down, and make appropriate corrections so that the code works for the test cases.

\*\* DO NOT REWRITE THE ENTIRE CODE. ONLY UPDATE THE PLACES WHERE YOU FIND AN ERROR.

\*\* DO NOT INCLUDE NEW MODULES OR EXTRA FUNCTIONS

def compute\_hcf(x, y):

if x > y:

s = y

else:

s = x

for i in range(1, s+1):

if((x % i == 1) and (y % i == 0)):

hcf = i

return hcf

num1 = int(input())

num2 = int(input())

print("The H.C.F. is", compute\_hcf(num1, num2))

54

24

The H.C.F. is 6

13

11

The H.C.F. is 1

100

50

The H.C.F. is 50

ANS:

def compute\_hcf(x, y):

if x > y:

s = y

else:

s = x

for i in range(1, s+1):

if((x % i == 0) and (y % i == 0)):

hcf = i

return hcf

num1 = int(input())

num2 = int(input())

print("The H.C.F. is", compute\_hcf(num1, num2))

Q4. The following is a program to print the factorial of a number taken as input. Copy-paste the code down, and make appropriate corrections so that the code works for the test cases.

\*\* DO NOT REWRITE THE ENTIRE CODE. ONLY UPDATE THE PLACES WHERE YOU FIND AN ERROR.

\*\* DO NOT INCLUDE NEW MODULES OR EXTRA FUNCTIONS

def recur\_factorial(n):

if n == 0:

return n

else:

return n+recur\_factorial(n-1)

num = int(input("Enter a number: "))

if num < 0:

print("Sfactorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of",num,"is",recur\_factorial(num))

Enter a number: 0

The factorial of 0 is 1

Enter a number: 1

The factorial of 1 is 1

Enter a number: 10

The factorial of 10 is 3628800

Enter a number: 15

The factorial of 15 is 1307674368000

Ans:

def recur\_factorial(n):

if n == 1:

return n

else:

return n\*recur\_factorial(n-1)

num = int(input("Enter a number: "))

if num < 0:

print("factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of",num,"is",recur\_factorial(num))

Q5. The following is a program to print the list of data Items appearing odd number of times in a List. Copy-paste the code down, and make appropriate corrections so that the code works for the test cases.

\*\* DO NOT REWRITE THE ENTIRE CODE. ONLY UPDATE THE PLACES WHERE YOU FIND AN ERROR.

\*\* DO NOT INCLUDE NEW MODULES OR EXTRA FUNCTIONS

x = eval(input())

l1 = []

for i in x:

if x.count(i) % 2 == 0:

if i in l1:

l1.append(i)

print(l1)

[1,2,3,4,5,1,3,3,4]

[2, 3, 5]

[1,2,3,4,4,4,4,4,5,5]

[1, 2, 3, 4]

[1,1,2,2,3,3,4,5,6,7,7,8,8,8,8,8,9]

[4, 5, 6, 8, 9]

x = eval(input())

l1 = []

for i in x:

if x.count(i) % 2 != 0:

if i not in l1:

l1.append(i)

print(l1)