PYTHON

Question 1: C) %

Question 2: B) 0

Question 3: C) 24

Question 4: A) 2

Question 5: D) 6

Question 6: C) the finally block will be executed no matter if the try block raises an error or not.

Question 7: A) It is used to raise an exception

Question 8: C) in defining a generator

Question 9: A) _abc and C) abc2

Question 10: A) yield and B) raise

Question 11:

```
# To take input from the user
num = int(input("Enter a number: "))
factorial=1


# To find the factorial of a number.
for i in range (1,num+1):
    factorial=factorial*i
print("The factorial of ",num,"is",factorial)
```

Question 12:

```
# To take input from the user
num = int(input("Enter a number: "))
```

```
# prime numbers are greater than 1
if num > 1:
   # check for factors
  for i in range(2,num):
     if (num % i) == 0:
        print(num,"is a composite number")
        print(i,"times",num//i,"is",num)
        break
  else:
     print(num,"is a prime number")

else:
  print(num,"is a composite number")
```

Question 13:

```
def isPalindrome(str):

   # Run loop from 0 to len/2
   for i in range(0, int(len(str)/2)):
      if str[i] != str[len(str)-i-1]:
         return False
   return True

# main function
s = (input("Enter the string: "))
ans = isPalindrome(s)
```

```
if (ans):
   print(s,"is a Palindrome")
else:
   print(s,"is not a Palindrome")
```

Question 14:

```
def rightangletriangle(opposite_side,adjacent_side,hypotenuse):
      if opposite_side == str("x"):
         return ("Opposite = " + str(((hypotenuse**2) - (adjacent_side**2))**0.5))
      elif adjacent_side == str("x"):
         return ("Adjacent = " + str(((hypotenuse**2) - (opposite_side**2))**0.5))
      elif hypotenuse == str("x"):
         return ("Hypotenuse = " + str(((opposite_side**2) +
(adjacent_side**2))**0.5))
      else:
         return "You know the answer!"


print(rightangletriangle(3,4,'x'))
print(rightangletriangle(3,'x',5))
print(rightangletriangle('x',4,5))
print(rightangletriangle(3,4,5))
```

Question 15:

```
# initializing string
test_str = "ShailzaFiroz"
```

```
# using set() + count() to get count
# of each element in string
res = {i : test_str.count(i) for i in set(test_str)}

# printing result
print ("The count of all characters in a given string is :\n "
                                        +  str(res))
```