Python Worksheet solution

1-c

2-B

3-C

4-D

5-D

6-C

7-A

8-C

9-A, C

10-B

11.
```
import re


def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)


# Taking input from the user
num_str = input("Enter a number: ")


# Using regular expression to check if the input is a non-negative integer
if re.match(r'^\d+$', num_str):
    num = int(num_str)
    result = factorial(num)
```

```
    print(f"The factorial of {num} is: {result}")
else:
    print("Invalid input. Please enter a non-negative integer.")
```

12-

```
import re

def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Taking input from the user
num_str = input("Enter a number: ")

# Using regular expression to check if the input is a non-negative integer
if re.match(r'^\d+$', num_str):
    num = int(num_str)
    if is_prime(num):
        print(f"{num} is a prime number.")
    else:
        print(f"{num} is a composite number.")
else:
    print("Invalid input. Please enter a non-negative integer.")
```

14--

```
import re

def find_third_side(side1, side2):
```

```
    if side1 < 0 or side2 < 0:
        return "Invalid input. Side lengths must be non-negative."
    else:
        third_side = (side1**2 + side2**2)**0.5
        return third_side

# Taking input from the user
side1_str = input("Enter the length of the first side: ")
side2_str = input("Enter the length of the second side: ")

# Using regular expressions to check if the input is a positive float or integer
if re.match(r'^\d+(\.\d+)?$', side1_str) and re.match(r'^\d+(\.\d+)?$', side2_str):
    side1 = float(side1_str)
    side2 = float(side2_str)

    third_side = find_third_side(side1, side2)

    if isinstance(third_side, str):
        print(third_side)
    else:
        print(f"The length of the third side is: {third_side}")
else:
    print("Invalid input. Please enter positive numbers for side lengths.")

15--import re

def character_frequency_regex(input_string):
    frequency = {}

    # Use a regular expression to match alphanumeric characters
    matches = re.findall(r'\w', input_string)
```

```
    for char in matches:
        frequency[char] = frequency.get(char, 0) + 1

    return frequency

# Taking input from the user
input_str = input("Enter a string: ")

# Calculating character frequencies using regular expression
result = character_frequency_regex(input_str)

# Displaying the results
print("Character frequencies:")
for char, count in result.items():
    print(f"'{char}': {count}")
```