Q1
Mark the following statements as true or false

1. The flow control statements are divided...

```
True
```

2. if -3 will evaluate to True

```
True
```

3. In Python, indented code block are always...

```
True
```

4. The finally block lets you execute code ....

```
True
```

5. The warning doesn’t stop the execution of a program ...

```
True
```

6. The ‘Pass’ statement in Python is used to raise an execption

```
False
```

7. The output of print(0.1 + 0.3 == 0.3) is True

```
False
```

8. The value of the expression 4 / (3 * (2-1)) and 4 / 3 * (2-1) is the same

```
False
```

9. A block begins when the line is indented by 4 spaces

```
Sure, you can do as many spaces as you want, as long as the all of them are in the same space
```

10. Python does have a switch statement ...

```
kind of. It’s match instead, but with small differences
```

11. In Python, you can write if-else statements in a single line ...

```
True
```

12. The pass statement is a null and can be used as a placeholder

```
True
```

14. The else clause in a try-execpt block is useful when you want to run a code when there is no execption

```
True
```

Q2
Choose the correct answer from the following multiple choices

1. These are the types of conditional statements

```
D
```

2. Print “Hello World” if a is greater than b

```
D
```

3. Print “Hello World” if a doesn’t equal b

```
D
```

4. Print “Yes” if a is equal to b, otherwise print “No”

```
A
```

5. Print “Hello” if a is equal to b, and c is euqal to d

```
A
```

6. Print “Hello” if a is equal to b, or if c is equal to d

```
B
```

7. Use the correct one line short hand syntax to print “Yes” ...

```
D
```

8. Use an if statement to print “Yes” if either a or b is equal to c

```
D
```

9. What is the output of the following Python code

```
C
```

10. Given the nested if-else below, what will be th value of x when the code executes successfully

```
D
```

11. What is the output of the following if statement

```
True
```

12. Given the nested if-else below, what is the ...

```
A
```

13. The except block lets you handle errors and the try block lets you test a block of code

```
B
```

14. You can use the .... keyword to define a block of code to be execuyted if no errors were raise

```
B
```

15. Python automatically generates many exception and errors...

```
D
```

16. Teh exception that occur at runtime in next python code is

```
A
```

Q
Conside the following Python code

1.

```
Division by Zero
Finally Clause is Executed
```

2.

```
A comma may be Missing in the Input
Finally Clause is Executed
```

3.

```
Result is 4.0
No Excecption
Finally Clause is Executed
```

4.

```
Something Wrong in the Input
Finally Clause is Executed
```

5.

```
Something Wrong in the Input
Finally Clause is Executed
```

6.

```
Something Wrong in the Input
Finally Clause is Executed
```

7.

```
Division by Zero
Finally Clause is Executed
```

Q
Complete the blank in next python code

1.

```
a = 50
b = 10

if a == b:
     print(“1”)
elif a > b:
     print(“2”)
else:
     print(“3”)
```

2.

```
if 5 > 2:
     print(“no way”)
```

Q
identify errors in the following Python code snapshot and then corret them

```
x = 10
if x > 5:
     print(“x is greater than 5”)
```

```
it should be
print(x, “is greater than 5”)
```

Q

Do these Programming Exercises with Python language to

1. Write a Python program to print whether a number is positive or negative using if-else

```
x = 5
if x % 2 == 0:
     print(“positive”)
else:
     print(“negative”)
```

2. Make a progrm that asks for a number between 1..10 ....

```
num = int(input(“Enter a number fom 1..10: ”))

if 0 > num > 10:
     print(“invalid number”)
else:
     print(“good”)
```

3. Make a program that asks for the password

```
password = input(“Enter your password: ”)
```

4. Implement a Python script to read a person age and display whether they are eligible for voting or not

```
age = int(input(“Enter your age: ”))

if age < 18:
     print(“still young”)
else:
     print(“nice”)
```

5. Implement a Python script to check whether the given year is a leap year or not

```
year = int(input("Enter the year: "))

if (year % 4 == 0) and (year % 100 != 0 or year % 400 != 0):
     print("leap")
else:
     print("no")
```

6. Check whether the user input is a positive or negative number

```
num = int(input("Enter the number: "))

if num < 0:
     print("negativef")
else:
     print("positive")
```

7. Write a program to find the largest number among three numbers

```
nums = []
for i in range(1, 4):
     num = input(f"Enter the number#{i}: ")
     nums.append(num)

max_num = nums[0]

for num in nums:
     if num > max_num:
          max_num = num

print("biggest is:", max_num)
```

8. Write a program to read a number and display the corresponsing day using if-else and else

```
```

```
    day_number = int(input("Enter a number (1-7) to get the
corresponding  day: "))

    if day_number == 1:
        print("Sunday")
    elif day_number == 2:
        print("Monday")
    elif day_number == 3:
        print("Tuesday")
    elif day_number == 4:
        print("Wednesday")
    elif day_number == 5:
        print("Thursday")
    elif day_number == 6:
        print("Friday")
    elif day_number == 7:
        print("Saturday")
    else:
        print("Invalid number! Please enter a number between 1 and
7.")
```

9. Whare are the optional statements possible inside a try-except block in Python

```
    I don’t understand, very confusing questions

    do you mean the else and finally keywords?, they’re optional
```

10. The else clause and the finally clause

```
    ```
    try:
        num = int(input("Enter a positive number: "))
        if num < 0:
            raise ValueError("Number must be positive.")
        print("Valid input:", num)
    except ValueError as e:
        print("Error:", e)
    else:
        print("No error, input was successful.")
    finally:
        print("This always runs - cleanup, logging, etc.")
    ```
```

11. Write a Python program to check wheter the given string is a plaindrome or not

```
s = input("Enter the text: ")
if s == s[::-1]:
     print("plaindrome")
else:
     print("not plaindrome")
```

12. Demonstrate a Python code to implement abnormal termination

```
int("hey")
```