CONDITIONAL STATEMENTS QUESTIONS

Section-A

1. What is a conditional statement in Python?
2. Explain the difference between the `if`, `elif`, and `else` statements.
3. How do you write a basic `if` statement in Python?
4. What is the purpose of indentation in Python conditional statements?
5. How do you use the `elif` statement in Python?
6. Explain the concept of nested `if` statements in Python.
7. What is the ternary conditional expression, and how is it used in Python?
8. How can you use the `pass` statement in an if block in Python?
9. Discuss the use of the `assert` statement in Python.
10. What is the purpose of the `if __name__ == "__main__":` block in Python scripts?
11. Explain the difference between the `==` and `is` operators in conditional statements.
12. How do you combine multiple conditions using logical operators (`and`, `or`) in Python?
13. Discuss the concept of truthy and falsy values in Python conditional statements.
14. How do you handle exceptions using the`try`, `except`, `else`, and `finally` blocks in Python?
15. Explain the use of the `in` and `not in` operators in Python conditional statements.
16. What is the purpose of the break and continue statements in loop structures within conditional statements?
17. How can you use the `assert` statement for debugging in Python?
18. Explain the concept of short-circuit evaluation in Python conditional statements.
19. How do you use the switch case statement in Python?
20. Discuss the differences between the `if` statement and the `if` expression (PEP 308) in Python.

Section-B

1. **What is Python?**
2. **What are the key features of Python?**
3. **What is PEP 8?**
4. **How is memory managed in Python?**
5. **Explain the difference between a list and a tuple.**

6. **What is the difference between shallow copy and deep copy in Python?**
7. **How do you handle exceptions in Python?**
8. **Explain the concept of generators in Python.**
9. **What is the purpose of "self" keyword in Python?**
10. **What is the difference between a module and a package in Python?**
11. **How do you comment on a single line and multiple lines in Python?**
12. **Explain the concept of indentation in Python.**
13. **How do you handle exceptions in Python? Give an example.**
14. **What is the purpose of the if…else statement in Python? Provide an example.**
15. **How do you define a function in Python? Give an example.**
16. **What is a module in Python? How do you import a module?**
17. **Explain the concept of class and an object in Python.**
18. **How do you create a file and write data into it using Python?**
19. **How do you read data from a file in Python?**
20. **Explain the concept of inheritance in Python.**
21. **What is the use of range () function in Python?**
22. **How do you write a conditional statement (if-else) in Python?**
23. **How do you handle exceptions in Python?**
24. **How do you read from and write to a file in Python?**
25. **What is object-oriented programming (OOP) in Python?**
26. **What is a class and how do you create an object in Python?**
27. **What is inheritance in Python? How does it work?**
28. **What are modules in Python? How do you import and use them?**
29. **How do you work with dates and time in Python?**
30. **What are some built-in libraries and functions in Python?**

Section-C

1. Describe Python's if-elif-else statement.
2. What is a for loop, and how does it work in Python?
3. Explain the while loop and give an example of its usage.
4. How do you write an if statement with multiple conditions in Python?
5. What is the purpose of the `else` clause in an if-elif-else statement?
6. Explain the concept of nested loops in Python.
7. How can you exit a loop prematurely in Python?
8. What is the `continue` statement used for in Python loops?
9. Describe the `range()` function in Python and its usage in for loops.
10. How do you iterate over the elements and their indices in a list using a for loop?
11. What is an infinite loop in Python, and how can you break out of it?
12. Explain the use of the `pass` statement in Python loops.
13. How do you iterate over the keys and values in a dictionary using a for loop?
14. What is the purpose of the `else` block in a while loop in Python?
15. How can you simulate a do-while loop in Python?
16. Explain the concept of a loop control variable.
17. What is the role of the `in` keyword in Python loops?
18. How do you find the maximum and minimum values in a list using loops inPython?
19. What are nested loops, and when are they commonly used?
20. Explain the use of the `range()` function with `for` loops in Python.
21. How can you avoid an infinite loop while waiting for user input in a Pythonprogram?
22. Describe the difference between `for` and `while` loops, and when should each beused?
23. How do you use the `enumerate()` function in a `for` loop in Python?
24. Explain the purpose of the `break` statement in Python loops.
25. How do you create a loop that iterates over a list in reverse order in Python?
26. What is the purpose of the `continue` statement in Python loops?
27. How can you iterate over characters in a string using a `for` loop in Python?
28. Explain the difference between a loop and a recursive function in Python.
29. What is an infinite loop, and how can you avoid it in Python?
30. How do you iterate over the elements of a list in reverse order using a `for` loop inPython?

31. Explain the concept of an iterator and give an example in Python.

32. How can you skip a specific iteration in a for loop without terminating the loop inPython?
33. Describe the `range()` function with three arguments in Python.
34. What is the purpose of the `pass` statement in Python loops?
35. How can you use a for loop to iterate over a dictionary's keys and values inPython?
36. What is the role of an iterable in a `for` loop, and how can you create a customiterable in Python?
37. Explain the purpose of the `else` block in a `for` loop in Python.
38. How can you use the `zip()` function in Python to iterate over multiple iterablessimultaneously?
39. Describe the `range()` function with one argument in Python.
40. How do you create an infinite loop in Python, and when might you use one?
41. What is the purpose of the `for-else` construct in Python?
42. How can you control the flow of a loop using the `continue` statement in Python?
43. What is the difference between a loop and a conditional statement in Python?
44. Explain the purpose of the `break` statement in Python loops.
45. How can you use the `while` loop to iterate over elements in a list in Python?
46. What is the `enumerate()` function in Python, and how is it used in loops?
47. Explain the use of the `in` operator in Python loops.
48. How do you find the sum of elements in a list using a `for` loop in Python?
49. Describe the `range()` function with two arguments in Python.
50. How do you create an infinite loop that terminates on user input in Python?

Section-D

Interview Questions on Python Control Flow(If Statements and Loops)

Q1. What is the primary purpose of the if statement in Python?

a) To execute a block of code based on a condition

b) To perform mathematical operations

c) To repeat a block of code

d) To define a function

Q2. What will be the output of the following code snippet?

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

a) x is greater than 5

b) x is less than 5

c) x is equal to 5

d) No output

Q3. Which keyword is used to execute a block of code if the condition in the if statement is false?

a) else

b) elif

c) while

d) for

Q4. What is the purpose of the elif statement in Python?

a) To execute a block of code if the previous conditions are false

b) To define a loop

c) To perform arithmetic operations

d) To exit from a loop

Q5. What will be the output of the following code snippet?

```
x = 5
if x < 3:
    print("x is less than 3")
elif x == 3:
    print("x is equal to 3")
```

```
else:
    print("x is greater than 3")
```

a) x is less than 3

b) x is equal to 3

c) x is greater than 3

d) No output

Q6. How can you execute multiple statements under a single if block in Python?

a) Separate statements with a semicolon

b) Indent the statements to the same level

c) Use the and keyword between statements

d) Use the elif keyword

Q7. What will the following code snippet print?

```
x = 10
if x < 5:
    print("x is less than 5")
elif x > 15:
```

```
    print("x is greater than 15")
else:
    print("x is between 5 and 15")
```

a) x is less than 5

b) x is greater than 15

c) x is between 5 and 15

d) No output

Q8. What is the purpose of the while loop in Python?

a) To execute a block of code repeatedly until a condition is false

b) To execute a block of code a fixed number of times

c) To define a function

d) To iterate over items in a sequence

Q9. What is the syntax for a while loop in Python?

a) while condition:

b) while condition():

c) while (condition):

d) while loop condition:

Q10. How can you exit a loop prematurely in Python?

a) Using the break statement

b) Using the continue statement

c) Using the pass statement

d) Using the exit function

Q11. What is the purpose of the for loop in Python?

a) To execute a block of code repeatedly until a condition is false

b) To iterate over items in a sequence

c) To execute a block of code a fixed number of times

d) To define a function

Q12. What is the syntax for a for loop in Python?

a) for item in sequence:

b) for item in range(n):

c) for index in range(len(sequence)):

d) All of the above

Q13. What will the following code snippet print?

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

a) apple banana cherry

b) [“apple”, “banana”, “cherry”]

c) 0 1 2

d) No output

Q14. How can you skip the current iteration of a loop and continue with the next iteration?

a) Using the skip statement

b) Using the pass statement

c) Using the break statement

d) Using the continue statement

Q15. What is the purpose of the range() function in Python?

a) To generate a sequence of numbers

b) To iterate over items in a sequence

c) To define a function

d) To execute a block of code repeatedly until a condition is false

Q16. What will the following code snippet print?

```
for i in range(3):
    print(i)
```

a) 0 1 2

b) 1 2 3

c) 2 1 0

d) 3 2 1 0

Q17. What will be the output of the following code?

```
for i in range(1, 6):
    if i == 3:
        continue
    print(i)
```

a) 1 2

b) 1 2 3

c) 1 2 4 5

d) 1 2 4

Q18. What will be the output of the following code?

```
for i in range(3):
    for j in range(3):
        print(i + j, end=' ')
    print()
```

a) 0 1 2

1 2 3

2 3 4

b) 0 1 2

1 2 3

2 3 4

3 4 5

c) 0 1 2

1 2 3

2 3 4

4 5 6

d) 0 1 2

2 3 4

4 5 6

Q19. What will be printed by the following code?

```
num = 5
while num > 0:
    print(num)
    num -= 1
    if num == 3:
        break
else:
    print("Done")
```

a) 5 4 3 2 1

b) 5 4

c) Done

d) 5 4 3

Q20. What is the output of the following code?

```
x = 10
if x > 5:
```

```
    print("A")
elif x > 7:
    print("B")
else:
    print("C")
```

a) A

b) B

c) C

d) A and B

Q21. What will be the output of the following code?

```
x = 5
while x > 0:
    print(x, end=" ")
    x -= 2
    if x == 1:
        break
else:
    print("Done")
```

a) 5 3 1

b) 5 3

c) 5 3 Done

d) 5 3 1 Done

Q22. What is the output of the following code?

```
for i in range(5):
    if i == 2:
        continue
    print(i, end=" ")
```

a) 0 1 3 4

b) 0 1 2 3 4

c) 0 1 3 4 5

d) 0 1 2 3 4 5

Q23. What is the output of the following code?

```
num = 0
while num < 5:
    print(num)
    num += 1
else:
    print("Loop completed.")
```

a) 0 1 2 3 4

b) 0 1 2 3 4 Loop completed.

c) Loop completed.

d) This code will result in an error.

Q24. What will be the output of the following code?

```
x = 10
if x > 5:
    print("Greater than 5")
if x > 8:
    print("Greater than 8")
if x > 12:
    print("Greater than 12")
else:
    print("Equal or less than 12")
```

a) Greater than 5

Greater than 8

Equal or less than 12

b) Greater than 5

Greater than 8

Greater than 12

Equal or less than 12

c) Greater than 5

Greater than 8

d) Equal or less than 12

Q25. What will be the output of the following code?

```
for i in range(3):
    for j in range(3):
        print(i * j, end=' ')
    print()
```

a) 0 0 0

0 1 2

0 2 4

b) 0 1 2

0 2 4

0 3 6

c) 0 0 0

0 1 2

0 2 4

0 3 6

d) 0 0 0

0 1 2

0 1 2

Q26. What will be the output of the following code?

```
num = 10
while num > 0:
    print(num)
    num -= 3
else:
    print("Loop completed.")
```

a) 10 7 4 1 Loop completed.

b) 10 7 4 1

c) Loop completed.

d) This code will result in an error.

Q27. What will the following code snippet print?

```
for i in range(3):
    pass
print(i)
```

a) 0 1 2

b) 1 2 3

c) 2 1 0

d) 3 2 1 0

Q28. What is the purpose of the `pass` statement in Python?

a) To exit from a loop prematurely

b) To skip the current iteration of a loop

c) To execute a block of code if a condition is false

d) To do nothing and act as a placeholder

Q29. What is the purpose of the else block in a loop in Python?

a) To execute if the loop encounters an error

b) To execute if the loop completes without encountering a break statement

c) To execute if the loop encounters a continue statement

d) To execute if the loop encounters a pass statement

Q30. What is the output of the following code?

```
x = 10
if x > 5:
    print("Hello")
elif x > 8:
    print("Hi")
else:
    print("Hey")
```

a) Hey

b) Hi

c) Hello

d) Hello, Hi

Q31. What is the output of the following code?

```
for i in range(1, 6):
    if i % 2 == 0:
        print(i)
        continue
    print("*")
```

a) \n2\n\n4\n*

b) \n2\n\n4\n*\n

c) \n2\n\n*\n4\n

d) \n\n2\n*\n4\n

Q32. What will be the output of the following code?

```
x = 5
while x > 0:
    print(x)
    x -= 1
else:
    print("Done")
```

a) 5\n4\n3\n2\n1\nDone

b) Done\n5\n4\n3\n2\n1

c) 5\n4\n3\n2\n1

d) Done

Q33. What will be the output of the following code?

```
for i in range(1, 6):
    if i == 3:
        break
    print(i)
else:
    print("Loop completed.")
```

a) 1\n2

b) 1\n2\n3\n4\n5\nLoop completed.

c) 1\n2\n3\n4\n5

d) 1\n2\n3

Section - E

1. Write a Python program to find those numbers which are divisible by 7 and multiples of 5, between 1500 and 2700 (both included).

2. Write a Python program to convert temperatures to and from Celsius and Fahrenheit.
[Formula : c/5 = f-32/9 [where c = temperature in celsius and f = temperature in

fahrenheit]

Expected Output :

60°C is 140 in Fahrenheit

45°F is 7 in Celsius

3. Write a Python program to guess a number between 1 and 9.

Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a "Well guessed!" message, and the program will exit.

4. Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

5. Write a Python program that accepts a word from the user and reverses it.

6. Write a Python program to count the number of even and odd numbers in a series of numbers

Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)

Expected Output :

Number of even numbers : 5

Number of odd numbers : 4

7. Write a Python program that prints each item and its corresponding type from the following list.

Sample List : datalist = [1452, 11.23, 1+2j, True, 'w3resource', (0, -1), [5, 12], {"class":'V', "section":'A'}]

8. Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.
Note : Use 'continue' statement.
Expected Output : 0 1 2 4 5

9. Write a Python program to get the Fibonacci series between 0 and 50.
Note : The Fibonacci Sequence is the series of numbers :
0, 1, 1, 2, 3, 5, 8, 13, 21, ...
Every next number is found by adding up the two numbers before it.
Expected Output : 1 1 2 3 5 8 13 21 34

10. Write a Python program that iterates the integers from 1 to 10 and 50 to 75 . For multiples of three print "Fizz" instead of the number and for multiples of five print "Buzz". For numbers that are multiples of three and five, print "FizzBuzz".
Sample Output :
fizzbuzz
1
2
fizz
4
buzz

11. Write a Python program that takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.
Note :
i = 0,1.., m-1
j = 0,1, n-1.

Test Data : Rows = 3, Columns = 4
Expected Result : [[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]
12. Write a Python program that accepts a sequence of lines (blank line to terminate) as input and prints the lines as output (all characters in lower case).

13. Write a Python program that accepts a sequence of comma separated 4 digit binary numbers as its input. The program will print the numbers that are divisible by 5 in a comma separated sequence.
Sample Data : 0100,0011,1010,1001,1100,1001
Expected Output : 1010

14. Write a Python program that accepts a string and calculates the number of digits and letters.
Sample Data : Python 3.2
Expected Output :
Letters 6
Digits 2

15. Write a Python program to check the validity of passwords input by users.
Validation :

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

16. Write a Python program to find numbers between 100 and 400 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

17. Write a Python program to print the alphabet pattern 'A'.
Expected Output:

```
***
 *  *
 *  *
```

```
 *****
 *    *
 *    *
 *    *
```

18. Write a Python program to print the alphabet pattern 'D'.

Expected Output:

```
****
 *    *
 *    *
 *    *
 *    *
 *    *
 ****
```

19. Write a Python program to print the alphabet pattern 'E'.

Expected Output:

```
*****
 *
 *
 ****
 *
 *
 *****
```

20. Write a Python program to print the alphabet pattern 'G'.

Expected Output:

```
***
 *    *
 *
 * ***
 *    *
 *    *
```

```
  ***
```

21. Write a Python program to print the alphabet pattern 'L'.

Expected Output:

```
*
 *
 *
 *
 *
 *
 *****
```

22. Write a Python program to print the alphabet pattern 'M'.

Expected Output:

```
*       *
  *       *
  * *   * *
  *   *   *
  *       *
  *       *
  *       *
```

Click me to see the sample solution

23. Write a Python program to print the alphabet pattern 'O'.

Expected Output:

```
***
 *   *
 *   *
 *   *
 *   *
 *   *
  ***
```

24. Write a Python program to print the alphabet pattern 'P'.

Expected Output:

```
****
 *   *
 *   *
 ****
 *
 *
 *
```

25. Write a Python program to print the alphabet pattern 'R'.

Expected Output:

```
****

 *   *
 *   *
 ****
 * *
 *  *
 *   *
```

26. Write a Python program to print the following patterns.

Expected Output:

```
****
 *
 *
  ***
     *
     *
 ****

 ooooooooooooooooo
ooooooooooooooooo
ooooooooooooooooo
```

```
oooo
oooo
oooo
oooooooooooooooo
oooooooooooooooo
oooooooooooooooo
                oooo
                oooo
                oooo
oooooooooooooooo
oooooooooooooooo
oooooooooooooooo
```

Click me to see the sample solution

27. Write a Python program to print the alphabet pattern 'T'.
Expected Output:

```
*****
  *
  *
  *
  *
  *
  *
```

28. Write a Python program to print the alphabet pattern 'U'.

Expected Output:

```
*   *
 *   *
 *   *
 *   *
 *   *
 *   *
  ***
```

29. Write a Python program to print the alphabet pattern 'X'.

Expected Output:

```
*   *
 *   *
  * *
   *
  * *
 *   *
 *   *
```

30. Write a Python program to print the alphabet pattern 'Z'.

Expected Output:

```
*******
     *
    *
   *
  *
 *
*******
```

31. Write a Python program to calculate a dog's age in dog years.

Note: For the first two years, a dog year is equal to 10.5 human years. After that, each dog year equals 4 human years.

Expected Output:

```
Input a dog's age in human years: 15
The dog's age in dog's years is 73
```

32. Write a Python program to check whether an alphabet is a vowel or consonant.

Expected Output:

```
Input a letter of the alphabet: k
k is a consonant.
```

33. Write a Python program to convert a month name to a number of days.

Expected Output:

```
List of months: January, February, March, April, May, June, July,
August
, September, October, November, December
Input the name of Month: February
No. of days: 28/29 days
```

34. Write a Python program to sum two integers. However, if the sum is between 15 and 20 it will return 20.

35. Write a Python program that checks whether a string represents an integer or not.

Expected Output:

```
Input a string: Python
The string is not an integer.
```

36. Write a Python program to check if a triangle is equilateral, isosceles or scalene.

Note :

An equilateral triangle is a triangle in which all three sides are equal.

A scalene triangle is a triangle that has three unequal sides.

An isosceles triangle is a triangle with (at least) two equal sides.

Expected Output:

```
Input lengths of the triangle sides:
x: 6
y: 8
z: 12
Scalene triangle
```

37. Write a Python program that reads two integers representing a month and day and prints the season for that month and day.

Expected Output:

```
Input the month (e.g. January, February etc.): july
Input the day: 31
Season is autumn
```

38. Write a Python program to display the astrological sign for a given date of birth.

Expected Output:

```
Input birthday: 15
Input month of birth (e.g. march, july etc): may
Your Astrological sign is : Taurus
```

Click me to see the sample solution

39. Write a Python program to display the sign of the Chinese Zodiac for the given year in which you were born.

Expected Output:

```
Input your birth year: 1973
Your Zodiac sign : Ox
```

40. Write a Python program to find the median of three values.

Expected Output:

```
Input first number: 15
Input second number: 26
Input third number: 29
The median is 26.0
```

41. Write a Python program to get the next day of a given date.

Expected Output:

```
Input  a year:  2016
Input a month [1-12]: 08
Input a day [1-31]: 23
The next date is [yyyy-mm-dd] 2016-8-24
```

42. Write a Python program to calculate the sum and average of n integer numbers (input from the user). Input 0 to finish.

43. Write a Python program to create the multiplication table (from 1 to 10) of a number.

Expected Output:

```
Input a number: 6
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
```

44. Write a Python program to construct the following pattern, using a nested loop number.

Expected Output:

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
```

45. Python program to check if a number is odd or even using if ststement.

46. Python program to check if a number is odd or even using if else .

47. Python program to check if a string matches the condition.

48. Python program to check if a number is odd and divisible by 3.

Section – F

1. Write a Python Program to Check weather a Given Number is odd or Even.
2. Write a Python Program to Check weather a candidate Eligible for Vote or not.
3. Write a Python program to check weather Given Number is Divisible by 7 or Not.
4.Write a program to display "Hello" if a number entered by user is a multiple of five , otherwise print "Bye".
5. Write a program to accept percentage from the user and display the grade according to the following **
**** criteria:**
** Marks Grade**
marks > 80: A+ grade
marks > 70 : A grade
marks > 60: B grade
** marks > 50: C grade**
else Failed.

6.Write a program that interprets the Body Mass Index (BMI) based on a user's weight and height.
It should tell them the interpretation of their BMI based on the BMI value.
• Under 18.5 they are underweight
• Over 18.5 but below 25 they have a normal weight
• Over 25 but below 30 they are slightly overweight
• Over 30 but below 35 they are obese
• Above 35 they are clinically obese.

The BMI is calculated by dividing a person's weight (in kg) by the square of their height (in m) .
Take height and weight from the user.