GATE

COMPLETE PRACTICE SHEET

Data Science & Artificial Intelligence & NIC - PARAM

Python-For Data Science

Q1 Which of the following statements is/are TRUE?

(A) In Python, a variable is used to store data values in memory.

(B) Mutable data objects in Python can be changed after they are created.

(C) Variables do not need to be declared with a specific data type.

(D) Python is a dynamically typed language, which means variables are untyped, and values have associated types.

Q2 Which of the following statements is/are INCORRECT?

(A) Python variables can be assigned new values of different types without any issues.

(B) In Python, "types" refer to the classification of data values that indicate the kind of data they hold, such as integers or strings.

(C) Python variable names are not case-sensitive, so " VAR1" and "var1" refer to the same variable.

(D) Using function names like "print" and "min" as variable names in Python is always recommended to avoid conflicts.

Q3 Consider the following code snippet, what will be the type of variable z?

x = 7

y = 5

z = x / y

(A) int

(B) float

(C) str

(D) bool

Q4 Which of the following variable names is not allowed in Phython?

(A) my_var (B) _private

(C) 123_var (D) Max_value

Q5 How does the round() function work in Python?

(A) round() always rounds up to the nearest integer value.

(B) round() rounds to the nearest integer value, always rounding up if the decimal part is greater than or equal to 0.5.

(C) round() rounds to the nearest integer value, always rounding down if the decimal part is greater than or equal to 0.5.

(D) round() rounds to the nearest integer value, but if the decimal part is exactly 0.5, it rounds to the nearest even integer.

Q6 Which of the following statements accurately describes the mutability of the int, float, and str data types?

(A) Mutable: int, float. Immutable : Str

(B) Mutable: int. Immutable: float str

(C) All three are immutable.

(D) All three are mutable.

Q7 What does it mean for a data type to be "immutable"?

(A) An immutable data type cannot be modified after it has been created. Any operation

GATE

that seems to modify it actually creates a new instance.

(B) An immutable data type can be changed after it has been created without creating a new instance.

(C) An immutable data type means that you can only ever have one variable of this type in your program.

(D) An immutable data type means that no variable can be of this type.

Q8 What is the relationship between the values "25" and 25?

(A) They are of the same type since they both represent the number 25.

(B) They are of different types: "25" is a string, and 25 is an integer.

(C) They have different types, but they are interchangeable because Python treats them as equivalent values.

(D) They're both funnier than 24.

Q9 How does the int() function work?

(A) int() always rounds up to the nearest integer value.

(B) int() rounds to the nearest integer value, always rounding up if the decimal part is greater than or equal to 0.5.

(C) int() truncates the decimal part of float and returns the whole number part.

(D) int() rounds to the nearest integer value, but if the decimal part is exactly 0.5, it rounds to the nearest even integer.

Q10 Consider the following python code and identify the output:

```
m = 4
n = 6
m, n = n, m
print(m, n)
```

Q11 Consider the following python code and identify the output:

```
print(15 // 3)
print(15 / 3)
```

Q12 Consider the following python code and identify the output:

```
print(5 * 5)
print(5.5 * 5)
```

Q13 Consider the following python code and identify the output:

```
x = 11
y = 2
result = x % y
print(result)
```

Q14 Consider the following python code and identify the output:

```
print(17 // 3 + 1.5)
```

Q15 Which of the following statements is/are TRUE?

(A) The input() function always reads what the user typed in as a string.

(B) Escape sequences in Python are used to represent special characters, and they always start with the backslash.

(C) The ord() function returns the ASCII code (number) of a character.

(D) The + and * operators can be used on strings.

Q16 Which of the following statements is/are INCORRECT?

(A) The format() function always generates an integer representation of whatever you give it.

(B) x ** y is different from pow(x, y).

(C) The chr() function returns the character corresponding to a given ASCII code.

(D)

GATE

To use functions like max() and min(), you do not need to import the math module.

Q17 Consider the following statements:
S1: To convert a letter from upper to lower, you can add 32 to its ASCII value.
S2: The print() and format() functions are interchangeable in all cases.
Which of the following is CORRECT?
(A) Only S1 is correct.
(B) Only S2 is correct.
(C) Both S1 and S2 are correct.
(D) Neither S1 nor S2 are correct.

Q18 What is a possible result from random ()?
(A) 0.5 (B) 0
(C) 1 (D) Any of A, B, C

Q19 Which of the following correctly match the escape sequences for tab, newline, and backslash?
(A) \tt, \new, \\\ (B) \t, \n, \\
(C) \tab, \n1, \\\ (D) \tab, \n, \\

Q20 What is the difference between the end and sep parameters in the print () function?
(A) end specifies the separator between successive arguments; sep specifies the string added at the end of the printed output.
(B) end specifies the separator between successive arguments; sep specifies the string added before each argument.
(C) sep specifies the separator between successive arguments; end specifies the string added at the end of the printed output.
(D) None of These

Q21 If we execute the code value = input() and the user enters 25, can we then perform the operation value2 = value + 5 without an error?
(A) Yes, it will work without any issues.
(B) No, it will result in a runtime error.
(C) Yes, but value2 will be "255".
(D) None of These

Q22 What is the result of the following code?
```
s1 = "pqr"
s2 = "def"
 result = s1 * 3 + s2
 print (result)
```
(A) "pqrpqrpqrdef"
(B) "s1s1s1s2"
(C) "pqr333def"
(D) Error

Q23 Write the output:
```
print (format (421.698, "0.if"))
```

Q24 Write the output:
```
print(max (–30, 10, 27, –3) + min(–30, 10, 27, –3)
```

Q25 Write the output:
```
letter = 'y'
print (chr (letter) – 6), chr (ord(letter) – 34))
```

Q26 Write the output:
```
 x = 6
 y = 11
 z = 16
print (y, x, z sep = " –", end = "!")
```

Q27 Write the output:
```
 print(format (20.2, ">5. 0f") + "\\\\")
```

Q28 Which of the following statements is/are TRUE?

(A) bool("True") evaluates to True.
(B) bool("False") evaluates to False.
(C) "True" and True are different types of objects.
(D)

GATE

The only possible Boolean values are True, False, and None.

Q29 Which of the following statements is/are TRUE?
(A) None and "None" are equivalent.
(B) The order of the conditions in an if-elif-else block does not matter.
(C) The elif clause can be used multiple times within an if-elif-else block.
(D) The not operator is used to reverse the logical state of its operand.

Q30 Which of the following statements is/are TRUE?
(A) Nested if statements are if statements that are placed inside each other, allowing for multiple levels of conditional execution.
(B) The elif clause can only appear in an if statement with an else clause.
(C) The else clause can be used outside an if statement.
(D) The = operator is used to compare two values for equality, while the == operator is used to assign a value to a variable.

Q31 Can a variable declared within an if statement be accessed outside of that if statement?
(A) Yes, it can.
(B) No, it cannot.
(C) It depends on whether the declaration executes.
(D) Only if the variable is defined as global.

Q32 What is the result of the logical operation True and False?
(A) True (B) False
(C) None (D) Error

Q33 Which logical operator in Python returns True if at least one of the operands is True?
(A) and (B) or
(C) not (D) is

Q34 When would you typically use a conditional expression in Python?
(A) When you need to execute a block of code repeatedly.
(B) When you need to iterate over a known sequence of elements.
(C) When you need to assign a value based on a condition.
(D) When you want to handle errors and exceptions.

Q35 What happens if none of the conditions in an if-elif-else block are met?
(A) The program terminates.
(B) The block of code associated with the else statement is executed.
(C) The program moves to the next line of code outside the if-elif-else block.
(D) An error is raised.

Q36 When would you typically use an elif clause instead of multiple if statements?
(A) When you want to provide a default block of code to execute when none of the previous conditions are met.
(B) When you have multiple conditions to check, but only one of them should be executed.
(C) When you need to execute a block of code repeatedly.
(D) When you want to handle specific cases within a larger set of conditions.

Q37 Consider the expression x >= 0 or y < 10. If we assume x is equal to 10, and y is equal to 5, then Python will evaluate.
(A) Python will evaluate both conditions in left-right order to determine whether the expression is True.
(B) Python will evaluate both conditions in right-left order to determine whether the expression is True.

GATE

(C) Python will evaluate only the second condition, y < 10.
(D) Python will evaluate only the first condition, x >= 0.

Q38 Which of the following statements is true?
(A) (x >= 5 and x < 15) is the same as (5 <= x < 15)
(B) (y > 0 or y < -10) is the same as y > 0 and y < -10
(C) (x > 0 or x < -5 and y < 0) is the same as ((x > 0 or x < -5) and y < 0)
(D) A and C are both true

Q39 Which of the following is equivalent to the expression
(p + q > r) and (s – t <= u)?
(A) (p + q > r) and not (s - t >= u)
(B) not (p + q < r) and (s – t <= u)
(C) (p – q <= r) and (t – s > u)
(D) not ((p + q <= r) or (s – t > u))

Q40 What does the expression 5 + 10 * 2 // 3 - 4 evaluate to?
(A) 6 (B) 7
(C) 8 (D) 9

Q41 Consider the given code and identify the output

```
x = 10
y = 3
z = (x ** 2) % y + x // y + y
print(z)
```

Q42 Consider the given code and identify the output

```
x = 3
y = 5
z = not ((x + y) == (x * y) and (x % y) == (y // x))
print(z)
```

Q43 Consider the given code and identify the output

```
x = 5
y = 10
z = 2
result = x and (x + y * z == 30) and (5 != True)
print(result)
```

Q44 Consider the given code and identify the output

```
gate = None
if gate == None and (bool(0.0) != bool(0)):
  print("data")
elif not gate or (gate == "None" and bool("artificialIntelligence")):
  print("science")
else:
  print("python")
```

Q45 Consider the given code and identify the output

```
python, java, program = 6, 10, 0
if python == java or not program:
  program = 1
elif java > python and not python:
  program = 2
if program == True:
  print("GATE2024")
else:
  print("GATE2025")
```

Q46 Consider the given code and identify the output

```
ruby, python, perl = 5, 6, -1
if (ruby == 5):
  perl += 3
elif (python == 6):
  perl = 4
if (python > 0):
  perl -= 1
if (ruby < 999):
  perl + 1
print(perl)
```

GATE

Q47 Which of the following statements is/are TRUE?

(A) The break statement can be used to exit a loop prematurely

(B) It is possible to use the break statement to exit multiple nested loops at once.

(C) The continue statement is used to skip the remaining code within the current iteration of a loop and move to the next iteration.

(D) The sequence generated from range(n), where n is an integer, starts at 1 and goes up to (but does not include) n.

Q48 Which of the following statements is/are TRUE?

(A) Nested loops require using both for and while loops.

(B) Infinite loops occur when the condition controlling the loop never becomes False or when there is no exit from loop body.

(C) The while loop in Python is always guaranteed to execute at least once.

(D) A while True loop will run indefinitely until a break statement is encountered.

Q49 Consider the following statements:

S1: The loop condition in a while loop is evaluated before each iteration.

S2: The loop control variable in loops like for x in range(2) can be accessed outside the loop.

Which of the following is CORRECT?

(A) Only S1 is correct.

(B) Only S2 is correct.

(C) Both S1 and S2 are correct.

(D) Neither S1 nor S2 are correct.

Q50 Which of the following statements about the range() function in Python is true?

(A) The range() function can generate sequences of both numbers and strings.

(B) The range() function provides an option for randomization in for loops.

(C) The range() function can generate both continuous and non-continuous sequences of numbers in ascending or descending order.

(D) The range() function can only be used with for loops and not with while loops.

Q51 When would you typically use a while loop instead of a for loop?

(A) When you need to iterate over a known sequence of elements.

(B) When the number of iterations is fixed and predetermined.

(C) When the loop requires an exit condition that is not based on the number of iterations.

(D) When you want to execute a block of code at least once.

Q52 What is the purpose of nested loops?

(A) Nested loops allow for more efficient memory utilization.

(B) Nested loops are used to handle errors and exceptions that may occur during loop execution.

(C) Nested loops enable complex patterns of iteration by placing loops inside each other.

(D) Nested loops have no purpose. You can use a single loop in any instance you use nested loops.

Q53 In a for loop statement like for x in range(3), can you modify the value of the loop variable x within the loop body to change the number of iterations?

(A) Yes, the loop variable x can be modified within the loop body to change the number of iterations.

(B) No, modifying the loop variable x directly within a for loop will not a affect the number

GATE

of iterations.

(C) Modifying the loop variable x will cause an error and terminate the loop.

(D) The loop variable x can only be modified indirectly using additional variables.

Q54 What is the main difference between loops and conditionals (if/elif/else) in programming?

(A) Loops are used to execute a block of code repeatedly, while conditionals are used to make decisions based on different conditions.

(B) Loops and conditionals are interchangeable and can be used interchangeably in any programming scenario.

(C) Conditionals cannot be nested inside other conditionals, but loops can be nested inside other loops.

(D) Conditionals are used to iterate over a sequence of values, while loops are used to check for specific conditions and execute code accordingly.

Q55 Consider the given code and identify the output

```
count = 1
while count <= 10:
if count % 3 == 0:
count += 2
continue
print(count, end=" ")
count += 1
```

Q56 Consider the given code and identify the output

```
for i in range(4):
    for j in range(i):
        print(i + j, end=" ")
```

Q57 Consider the given code and identify the output

```
count = 0
while count < 5:
    print(count, end=" ")
    if count == 2:
        break
count += 1
```

Q58 Consider the given code and identify the output

```
for i in range(1, 10, 2):
    print(i, end=" ")
```

Q59 Consider the given code and identify the output

```
num = 10
while num > 0:
    if num % 2 == 0:
        num -= 1
else:
        num += 1
print(num, end=" ")
```

Q60 Consider the given code and identify the output

```
my_sum = 0
for i in range(1, 5):
    my_sum += i
print(i)
```

Q61 Consider the given code and identify the output

```
x = 10
while x in range(10, 5, -2):
    print("Python", end = " ")
    x -= 4
```

Q62 Which of the following statements is/are TRUE?

(A) A function must always have a return statement.

(B) A variable defined within a function is limited to that function and is not accessible

outside of it.

(C) You can call a function without passing any arguments, even if it has required parameters.

(D) When a function lacks a return statement or includes a return statement without specifying a value, it implicitly returns 0.

Q63 Which of the following statements is/are TRUE?

(A) A function can have multiple parameters with the same name as long as they have different default values.

(B) Keyword arguments are passed to a function based on their order in the function's parameters, while positional arguments are explicitly specified by name in the function call.

(C) When you pass a mutable object as an argument to a function in Python, the function can modify the object.

(D) A break statement and a return statement can be used interchangeably.

Q64 Consider the following statements:

S1: Functions can have multiple return statements, but only one of them will be executed during the function's execution.

S2: Functions can return multiple values using the return statement followed by a comma-separated list of values.

Which of the following is CORRECT?

(A) Only S1 is correct.

(B) Only S2 is correct.

(C) Both S1 and S2 are correct.

(D) Neither S1 nor S2 are correct.

Q65 What is the primary difference between using return and print()?

(A) Both return and print() serve the same purpose, so you can use either one interchangeably.

(B) return is used to exit a function and pass a value back to the caller, while print() is used to display output on the console.

(C) print() is used to pass values to the caller, while return is used to display output on the console.

(D) return and print() are used for the same purpose, but return is used when working with numerical values, and print() is used for strings.

Q66 Which of the following is not an advantage of using a function?

(A) Reusability of code.

(B) Easier debugging and maintenance.

(C) Increased program execution speed.

(D) Improved code organization and readability.

Q67 What is a requirement for a function?

(A) A function must have at least one parameter.

(B) A function must contain at least one return statement.

(C) A function must be defined using the def keyword.

(D) A function must have default values for its parameters.

Q68 What is the significance of proper indentation in relation to functions?

(A) Indentation within a function is optional and doesn't affect the function's behavior.

(B) Indentation is used to group code together and define the body of the function.

(C) Indentation is used to define the end of a function.

(D) Indentation is necessary to declare function parameters.

Q69 In which of the following scenarios would you not want to use default arguments for a

GATE

function?

(A) When you want to simplify function calls by allowing some arguments to be omitted.

(B) When you want to minimize potential confusion and ensure that function behavior is explicit.

(C) When you need the function to handle different cases with varying argument values.

(D) When you want to ensure that all function arguments are explicitly provided by the caller.

Q70 Consider the following code, and provide the output

```
def ruby(javascript, perl, python):
    return python + javascript / perl
language = ruby(perl = 3, python = 2, javascript = 6)
print(language)
```

Q71 Consider the following code, and provide the output

```
def language(hindi = 7):
    hindi *= 2
    return hindi * 2
javaDot = language("ruby!")
python = language(language())
print(javaDot, python)
```

Q72 Consider the following code, and provide the output

```
def color(grey):
    kitkat = True
    munch = False
    while not munch and kitkat:
        break
        return grey + 5
print(color(10))
```

Q73 Consider the following code, and provide the output

```
def data(work, study, eat, dance):
    return work + study * eat * dance
print(data(work = 5, dance = 2, study = 2, 3))
```

Q74 Consider the following code, and provide the output

```
def gate():
    dataScience = "python"
    COMPScience = "dataStructure"
        return COMPScience
        return dataScience
myFavoriteSubject = gate() # GO CRACK GATE!!!
print(myFavoriteSubject)
```

Q75 Consider the following code, and provide the output

```
def friend():
    return "ABC", "XYZ"
person1, person2 = friend()
print(person1, "< 3", person2)
```

Q76 Consider the following code, and provide the output

```
def countMarks():
    count = 0
    for num in range(10):
        count += 1
print(count)
```

Q77 Which of the following statements is/are TRUE?

(A) Characters are a different type from strings in Python

(B) If strings s1 and s2 are equivalent, the expression s1 in s2 will return False because "in" checks for partial matches and not the complete operand length.

(C) In Python, you can directly modify a character within a string by using the index notation, like s[2] = "z".

GATE

(D) To loop over the indices of string s, you can do for i in range(len(s) + 1).

Q78 Which of the following statements is/are TRUE?
(A) Strings in Python are 0-based indexed (i.e., the first character is at index 0).
(B) The strip() method in Python removes both leading and trailing whitespace from a string.
(C) Python allows you to compare strings using the relational operators (<, <=, >, >=) based on their lengths.
(D) Strings can indexed using negative numbers.

Q79 Consider the following statements:
S1: On string s, min(s) == max(s) will always evaluate to False.
S2: islower(), isdigit() and isalpha() are all functions, so they take a string as a parameter rather than being directly called on a string.
S3: When s is a string, in loops like for i in s, i will refer to a character of s each iteration.
How many of the above statement is/are CORRECT?

Q80 What is the result of "ink ink" * 2 + " ink"?
(A) "ink inkink ink ink"
(B) "ink inkink"
(C) "ink ink ink ink"
(D) None of the above

Q81 What is the result of "Feebee".find("e")?
(A) 1 (B) 2
(C) 4 (D) 5

Q82 Which operator can be used to check if a substring is present in a string?
(A) contains (B) in
(C) exists (D) substring

Q83 If s = "Padhle and Beta =)", what is the result of the expression len(s) - len(s.replace(" ", ""))?
(A) 0
(B) 1
(C) 2
(D) None of the above

Q84 Which of the following statements is true about Python strings?
(A) Strings can only contain letters and numbers.
(B) Strings can be modified in place.
(C) Strings must always be enclosed in double quotes.
(D) Strings can be converted to other data types.

Q85 What does the replace() method do when applied to a string?
(A) It removes all whitespace from the string.
(B) It replaces all occurrences of one substring with another.
(C) It reverses the characters in the string.
(D) It raises an error since strings are immutable.

Q86 If you want to traverse through a string S using negative indices in Python, which loop should you use?
(A) for i in range(-1, -len(S)-1, -1)
(B) for i in range(len(S))
(C) for i in range(-len(S), 0, -1)
(D) for i in range(-len(S), 0)

Q87 If s is a string, which expression will always evaluate to True?
(A) chr(ord(s))) == s
(B) s.upper().lower() == s
(C) s[0:len(s)] == s
(D) All of the above

Q88 Write the output of the following code:
s = "PythonPadlo"

GATE

```
new = ""
for char in s:
    new += chr(ord(char) - 3)
print(new)
```

Q89 Write the output of the following code:

```
s = "SillyPython-561"
result = ""
for char in s:
    if char in str(not s):
        result += "B"
    elif char.isalpha():
        result += char.upper()
    elif char.isdigit():
        result += str(int(char) + 3)
else:
        result += char
print(result)
```

Q90 Write the output for the following code:

```
s = "ShakalakaBoom"
result = (s[-len(s):-len(s)+5] + " " + s[-6:])
print(result)
```

Q91 Write the output of the following code

```
javascript = "doobyD00by"
python = "hurrah732h"
ruby = "+25"

pankaj = python.islower() and python[999:9999]
== ""
neeraj = ruby.isalnum() and ruby.isdigit()
kamal = javascript[6:8].isdigit() and
javascript.count("o") == 2

print(pankaj, Neeraj, kamal)
```

Q92 Write the output fo the following code:

```
def gateExam(subject1, subject2):
        subject1 = subject1 + subject2
        subject2 = subject1
        print(subject1 is subject2)
string1 = "database"
string2 = "python"
gateExam(string1, string2)
print(string1, string2)
```

Q93 Write the output of the following code:

```
myGateExam = "DataScience" # =)
print(myGateExam[0:1000] * 2)
```

Q94 Write the output of the following code:

```
alpha, beta = "cast", "dean"
omega = beta[:2] + alpha[2:]
sigma = beta[:len(beta)] + alpha[:3]
print(min(omega, sigma), max(omega, sigma))
```

Q95 Which of the following statements is/are TRUE?

(A) Comparing two lists of different lengths will result in an error.

(B) It is possible to directly convert a list to a string in Python

(C) The append() method adds an element to the beginning of a list in Python.

(D) The expression [x for x in range(10) if not x % 2] creates a list containing only positive even integers.

Q96 Which of the following statements is/are TRUE?

(A) Lists cannot be negatively indexed.

(B) The remove() method deletes an element at a specified index from a list.

(C) It is possible to convert items of other types directly to a list in Python.

(D) Lists can only contain items of the same data type

Q97 Consider the following statements:

S1: The += operator in Python can be used to concatenate two lists.

GATE

S2: Lists have a fixed maximum size, and once this size is reached, no more elements can be added to the list.

S3: The pop() method in Python is used to remove and return the last element of a list.

How many of the above statement is/are CORRECT?

Q98 What is the primary difference between the list methods .append() and .extend()?

(A) .append() is used to add a single element to the end of the list, while .extend() is used to add multiple elements to the end of the list.

(B) .extend() is used to add a single element to the end of the list, while .append() is used to add multiple elements to the end of the list.

(C) .append() is used for merging two lists, while .extend() is used to add a single element to a list.

(D) None of These

Q99 Which of the following list comprehensions is not valid in Python?

(A) [x for x in range(0)]

(B) [x**2 for x in range(5) if x % 2 == 0]

(C) [len(item) for item in ["my", "anaconda", "don't"]]

(D) [x if x > 0 else -1 for x in range(5)]

Q100 Assuming list1 is a list, which of the following declarations will not create a new list object with the same items as list1?

(A) list2 = list1[0:len(list1)]

(B) list3 = list1

(C) list4 = [goodies for goodies in list1]

(D) list5 = list(list1)

Q101 Assuming myChocolate = [“Dairymilk", "Kookie", "KitKat", “Turbo"], what does the ex-pression myChocolate[:-2] evaluate to?

(A) [“Dairymilk, "Kookie"]

(B) [“Dairymilk, "Kookie", "KitKat"]

(C) ["Kookie, "KitKat", “Turbo"]

(D) Error

Q102 In Python, when comparing two lists using the greater-than (>) or less-than (<) operators, how is the comparison evaluated?

(A) The comparison checks if the lengths of the lists are greater than or less than each other.

(B) The comparison checks if the elements of the lists, when converted to strings, are greater than or less than each other in lexicographical order.

(C) The comparison checks if the elements at corresponding indices in the lists are greater than or less than each other.

(D) List comparison using > or < is not allowed in Python.

Q103 What is the difference between the .find() and .index() methods in Python?

(A) .find() works only on strings and returns -1 if the specified element is not found, while .index() works on strings and lists but raises an error if the specified element is not found

(B) .find() only works on lists, and .index() only works on strings, but both return -1 if the specified element is not found.

(C) .find() and .index() are interchangeable methods, and there is no significant difference between them.

(D) .find() works on strings and lists, while .index() works only on strings, but
both raise an error if the specified element is not found.

Q104 Which of the following statements accurately describes the uses of list()?

(A) list() is used to create an empty list.

(B) list() can convert a string into a list, where each character becomes an element

GATE

(C) list() can convert other iterable objects like a range (from range()) into lists.
(D) All of the above.

Q105 Write the output of the following code:

```
school = [“pankaj", “neeraj", “kamal"]
school2 = school
school2.append(“fill")
print(school)
```

Q106 Write the output of the following code:

```
doubleTrouble = [“july", “john", "tom", "jerry"]
doubleTrouble = doubleTrouble.sort()
print(doubleTrouble)
```

Q107 Write the output of the following code:

```
playground = []
clubhouse = [“hockey", “football", “cricket", “tennis", “badminton", “cycling"]
for i in range(len(clubhouse)):
    playground += clubhouse[i]
print(playground)
```

Q108 Write the output of the following code:

```
white = [“himmat", “saahas", “budhiman"]
myFavoriteColor = [“himmat", “saahas", “buddhiman"]
print(myFavoriteColor is white and white == myFavoriteColor)
```

Q109 Write the output of the following code:

```
answer = [“padhlo", “bacho", “rank", “aa jayegi", 67]
answer.sort()
print(answer)
```

Q110 Write the output of the following code:

```
myNums = [1, 2, 3, 4, 5]
print([num ** 2 for num in myNums])
```

Q111 Write the output of the following code:

```
super = ["python"]
name = [“rohit", “dev", “aadi"]
for food in range(len(super)):
    super.append(name[food])
print(super)
```

Q112 Write the output of the following code:

```
lst1 = [“Pankaj", “Neeraj"]; lst2 = [“PadhloBeta", “GateNikalanaHaiTo"]
print(min(max(lst1, lst2)))
```

Q113 Write the output of the following code:

```
def iChooseYou(pc):
    pc.append(“Kamal")
    pc.sort()
faculty = [“Pankaj", “Neeraj"]
iChooseYou(faculty)
print(faculty)
```

Q114 Which of the following statements is/are TRUE?
(A) The expression myList[2][3] is valid for a nested list with three inner lists, each having at least four elements.
(B) In a linear search, if the specified item is found, the search algorithm returns the index of the found item plus 1 to account for 0-based indexing.
(C) Iterating through a nested list requires nested loops to access each individual element.
(D) Binary search is guaranteed to return the index of the first instance of an item in a list.

Q115 Which of the following statements is/are TRUE?
(A) Lists within a list can have a different length from each other.
(B) In selection sort, the minimum element is repeatedly selected from the unsorted part of the list and swapped with the first element, while in insertion sort, elements are

GATE

compared and inserted into their correct positions in the sorted part of the list.

(C) The len() function can be used to find the total number of elements in a nested list.

(D) When using the .sort() method on nested lists in Python, the sorting is applied only to the outermost list, leaving the inner lists unaltered.

Q116 Consider the following statements:

S1: When using the .sort() method on nested lists in Python, the sorting is applied only to the outermost list, leaving the inner lists unaltered.

S2: In a linear search, one item is checked in the unexplored portion of the list each step, while in binary search, the search space is cut in half with each step.

S3: In binary search, the number of "searches" required is at most half the length of the list.

How many of the above statement is/are CORRECT?

Q117 In which scenario would using binary search be more advantageous than linear search?

(A) Searching through a small, unsorted list.

(B) Looking for the first occurrence of an item in a list.

(C) Searching through a large, sorted list.

(D) None of these

Q118 How many iterations does a linear search require to find the value 616 in the list [9, 10, 21, 41, 98, 123, 364, 616, 1218]?

(A) 8 (B) 7

(C) 6 (D) 9

Q119 How many iterations does a binary search require to find the value 616 in the list [8, 11, 22, 43, 88, 133, 364, 616, 1212]?

(A) 1 (B) 2

(C) 3 (D) 4

Q120 What will the list [86485, 42, 1337, 404, 777, 9000, 24601] look like after three iterations of the selection sort algorithm?

(A) [42, 404, 777, 1337, 9000, 24601, 86485]

(B) [42, 404, 777, 86485, 1337, 9000, 24601]

(C) [42, 404, 1337, 86485, 777, 9000, 24601]

(D) None of the above.

Q121 What will the list [86485, 42, 1337, 404, 777, 9000, 24601] look like after three iterations of the insertion sort algorithm?

(A) [42, 404, 777, 1337, 9000, 24601, 86485]

(B) [42, 1337, 86485, 404, 777, 9000, 24601]

(C) [42, 404, 1337, 86485, 777, 9000, 24601]

(D) None of the above.

Q122 How do you remove "smart" from the following nested list?:

100features = [["cute", "intelligent", "mad"], ["handsome", "quick", "clever", "smart"], ["beautiful", "happy", "sad"]]

(A) 100features.remove("smart")

(B) 100features.pop(1)

(C) 100features [1].remove("smart")

(D) 100features [1].pop("smart")

Q123 How do you check if the string "stu" is present in the following nested list?:

myTeam = [["pankaj", "neeraj"], ["stu", "student"], ["shekhar", "kamal"]]

(A) "stu" in x

(B) True if "stu" in [item for item in x] else False

(C) "stu" in x[1]

(D) "stu" in x[1:1]

Q124 On a list that is already mostly sorted, which sorting algorithm is likely to perform worse, selection sort or insertion sort?

(A)

GATE

Insertion sort, because it will cause unneeded swaps as it sorts through the mostly sorted list.

(B) Selection sort, because it will unnecessarily search for minimum elements, which are likely to be in the correct place already in a mostly sorted list.

(C) The two algorithms will perform exactly the same.

(D) This cannot be determined from the given information.

Q125 Write the output of the following code:

```
queensLists = [[“ranilaxmibai"], [“ranidurgavati", “tarabai"], \
                                [“raziasultan", “padmavati", “gauradevi"]]
result = [homegirl for lst in queensLists for homegirl in lst]
print(result)
```

Q126 Write the output of the following code:

```
inputNums = [13, 17, 25, 28, 10, 30, 21]
myLst = [ [0, 0, 0], [0, 0, 0], [0, 0, 0] ]
for num in inputNums:
    result1 = num % 3
    result2 = (num // 10) % 3
    myLst[result1][result2] += 1
print(myLst)
```

Q127 Write the output of the following code:

```
powerful = [[“hosiyar", “damdaar"], \
  [“budhiman", “khatarnak"], \
  [“darpok", “shaktiman"]]
powerful.sort()
print(powerful)
```

Q128 Write the output of the following code:

```
lst1 = [[1, -100], [5, 10]]; lst2 = [[1, -900], [999, 999]]
print(lst1 > lst2)
```

Q129 Write the output of the following code:

```
gate = [["H", "X", "Z", "C"], ["A", "O", "T", "J"], \
          ["K", "P", "N", "D"], ["L", "M", "U", "K"]]
for i in range(len(gate)):
   for j in range(len(gate)):
      if i == j:
         print(gate[i][j], end = "")
```

Q130 Write the output of the following code:

```
subject = [[“python", “warehousing"], \
    [“datascience", “mathematics", “algorithms"], \
    [“aptitude", “networking"]
teacher = [group[0] for group in subject]
print(teacher)
```

Q131 Write the output of the following code:

```
faculty = [[“pankaj", “neeraj"], [1930, 4], [“kamal", “shweta", “anjali"]]
print(facuty[-1][-2][0])
```

Q132 Write the output of the following code:

```
myNums = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
total = 0
for i in range(len(myNums)):
   for j in range(4):
      if not j % 2:
         total += myNums[i][j]
print(total)
```

Q133 Which of the following statements is/are TRUE?

(A) Recursion is a programming technique where a function calls itself.

(B) A recursive function must have a base case to stop the recursion.

(C) Any recursive function using a list can be adjusted to use a set.

(D) Recursive functions are always more efficient than iterative solutions.

Q134 Which of the following statements is/are TRUE?

GATE

(A) Recursive functions can be called with different parameters in each recursive call.
(B) A recursive function can have multiple base cases.
(C) Recursive functions can only call themselves once within their body.
(D) Recursion is the only way to implement functions that solve certain mathematical problems efficient, such as factorials.

Q135 Which of the following statements is/are TRUE?
(A) If function A calls function B, and function B calls function A, then neither function is recursive because they do not directly call themselves.
(B) Recursion always leads to infinite recursive calls if not implemented correctly.
(C) Recursion is generally recommended for problems that can be easily solved using loops.
(D) Recursive functions in Python cannot be optimized for better performance.

Q136 What is the purpose of a base case in a recursive function?
(A) To make the code more readable and maintainable.
(B) To handle errors and exceptions that may occur during recursion.
(C) To define the initial condition that stops the recursive calls.
(D) To ensure that the function returns a value at every step of the recursion.

Q137 Which of the following is true about the relationship between recursive and iterative solutions?
(A) Recursive solutions are always clearer than iterative solutions.
(B) Iterative solutions are always more efficient than iterative solutions because the function does not call itself repeatedly.
(C) Iterative solutions are prone to errors such as infinite loops, while recursive solutions are not.
(D) Recursion and iteration are different techniques with their own strengths and weaknesses, although they can be used interchangeably.

Q138 What is the primary advantage of using recursion in programming?
(A) Recursion allows for more efficient memory utilization.
(B) Recursive implementations are always simpler because they do not use loops.
(C) Recursion can solve complex problems concisely by breaking them down into smaller instances.
(D) Recursion improves the speed of program execution.
(E) Recursive solutions are easier to debug.

Q139 What is a potential drawback of using recursion in programming?
(A) Recursion can lead to infinite function calls and stack overflow errors.
(B) Recursion is only applicable to mathematical calculations.
(C) Recursion can be challenging with some data structures, like dictionaries.
(D) Using recursion makes the code more difficult to read and understand.

Q140 Which of the following is a correct recursive implementation of the power function?
(A)
```
def power(x, n):
    if n == 0:
return 1
    return x * power(x, n - 1)
```
(B)
```
def power(x, n):
    if n == 1:
```

GATE

```
        return x
    else:
      return x * power(x, n - 1)
```

(C)
```
def power(x, n):
    return x ** n
```

(D)
```
def power(x, n):
    if n == 0:
        return 1
    else:
        return power(x, n) * x
```

Q141 What does the term "recursion depth" refer to?

(A) The number of recursive calls made by a function.

(B) The depth of nested loops in a recursive algorithm.

(C) The total number of lines in a recursive function.

(D) The level of indentation in a recursive function.

Q142 Under what circumstances is it opportune to use recursion in programming?

(A) When the problem can be naturally divided into smaller instances of the same problem, and requires a parameter that can be minimized or truncated

(B) Only when the problem is impossible or too complex to solve iteratively.

(C) When aiming for the most memory efficient solution due to the inherent nature of recursive algorithms.

(D) Whenever recursion is available as an option, as it often simplifies code and enhances maintainability.

Q143 What is the correct recursive implementation for a function that takes two ordered strings as input and returns their (ascending) ordered concatenation?

(A)
```
def ordered_concat(str1, str2):
    if not str1:
        return str2
    elif not str2:
        return str1
    else:
        return ordered_concat(str1[1:],
        str2[1:]) + max(str1[0], str2[0])
```

(B)
```
def ordered_concat(str1, str2):
    if len(str1) == 0:
        return str2
    elif len(str2) == 0:
        return str1
    else:
        return ordered_concat(str1[1:],
        str2[1:]) + min(str1[0], str2[0])
```

(C)
```
def ordered_concat(str1, str2):
    if not str1:
        return str2
    elif not str2:
        return str1
    elif str1[0] < str2[0]:
        return str1[0] + ordered_concat
        (str1[1:], str2)
    else:
        return str2[0] + ordered_concat
        (str1, str2[1:])
```

(D)
```
def ordered_concat(str1, str2):
    if len(str1) == 0:
        return str2
    elif len(str2) == 0:
        return str1
    elif str1[0] > str2[0]:
        return max(str1[0], str2[0]) +
    ordered_concat
        (str1[1:], str2)
    else:
        return str2[0] + ordered_concat
        (str1, str2[1:])
```

Q144 Write the output of the following code:

GATE

```
def lanaguage(python):
    if python <= 0:
        return
    print(python, end = " ")
    language(python - 2)
    print(python, end = " ")
print(language(5))
```

Q145 Write the output of the following code:

```
def padhlo(beta):
    if not beta:
        return 0
    elif beta[0] in "aeiouAEIOU":
        return 1 + padhlo(beta[1:])
else:
        return padhlo(beta[1:])
print(padhlo("RankLaoge"))
```

Q146 Write the output of the following code:

```
def language(python):
    if len(python) == 0:
        return ""
  else:
      return language(python[1:]) + python[0]
print(language("gate24exam"))
```

Q147 Write the output of the following code:

```
def subject(java, python):
if python == 0:
    return java
else:
    return subject(python, java % python)
print(subejct(15, 10), subejct(12, 18), subejct(49, 77))
```

Q148 Write the output of the following code:

```
def gupchup(gang):
    clues = []
    for who in gang:
        if type(who) == list:
            clues.extend(gupchup(who))
        else:
            clues.append(who)
    return clues
inc = ["pankaj", ["neeraj", ["shweta", "anjali"]], ["kiran"]]
print(gupchup(inc))
```

Q149 Write the output of the following code:

```
def girlPower( arr, l, r, x):
    if r < l:
        return -1
    if arr[l] == x:
        return l
    if arr[r] == x:
        return r
    return girlPower(arr, l+1, r-1, x)
    princesses = ["anjali", "himani", "kiran", "jasmine", "arpita", \
    "bhumi", "saumya", "tanuja"]
print(girlPower(princesses, 0, len(princesses) - 1, princesses[3]))
```

Q150 Write the output of the following code:

```
def trickyPython(scarySnake):
    if scarySnake == []:
        return 0
    elif scarySnake[0] % 2 == 0:
        return -scarySnake[0] + trickyPython(scarySnake[1:])
    else:
        return scarySnake[0] + trickyPython(scarySnake[1:])
    scarySnake = [7, -42, 17, -99, 2, -8]
    print(trickyPython(scarySnake))
```

Q151 Write the output of the following code:

```
def gateExam(padhlo, beta):
    if beta == 0:
        return 1
    elif beta % 2 == 0:
        ranker = gateExam(padhlo, beta // 2)
```

GATE

```
        return ranker * ranker
else:
        return padhlo * gateExam(padhlo beta - 1)
print(gateExam(3, 4), gateExam(4, 3))
```

Android App | iOS App | PW Website