



तत् त्वं पुष्पन् अपावृणु
केन्द्रीय विद्यालय संगठन

केन्द्रीय विद्यालय संगठन अहमदाबाद संभाग

KENDRIYA VIDYALAYA SANGATHAN
AHMEDABAD REGION



QUESTION BANK

CLASS – XII

SUBJECT – COMPUTER SCIENCE (083)

»» SESSION 2026-27 ««



THINK
Creatively



CODE
Confidently



SOLVE
Smartly

संरक्षक PATRONS



श्री धर्मेन्द्र पटले
माननीय उपायुक्त , के. वि. संगठन, अहमदाबाद संभाग
Shri Dharmendra Patle

HONOURABLE DEPUTY COMMISSIONER, K. V. SANGATHAN AHMEDABAD REGION



श्री वेंकटेश्वर प्रसाद बी.
Sh. Venkteswar Prasad B.



श्रीमती मीना जोशी
Smt. Meena Joshi



श्रीमती बिबियाना एक्का
Smt. Bibiyana Ekka

माननीय सहायक आयुक्त, के. वि. संगठन, अहमदाबाद संभाग
HONOURABLE ASSISTANT COMMISSIONER, K. V. SANGATHAN AHMEDABAD REGION



श्री देवेन्द्र सिंह शेखावत
पाठ्यक्रम निर्देशक एवं कार्यस्थल प्राचार्य
पीएम श्री केंद्रीय विद्यालय क्रमांक 3 वायुसेना स्थल - II, जामनगर
Sh. Devendra Singh Shekhawat
Course Director & Venue Principal
PM SHRI KENDRIYA VIDYALAYA NO.3 AIR FORCE II, JAMNAGAR



CONTENT CREATION TEAM

*Ideas Today...
Knowledge Tomorrow...*

*Create
Collaborate
Contribute
for a Better
Learning World*



Mr. Santosh Kumar Deval, PM SHRI KV Bhavnagar Para



Mr. Bharat Katariya, PMSHRI KV AFS Wadsar



Mr. A. S. Modi, PMSHRI KV Sec.30 Gandhinagar



Ms. Renu Baheti, PMSHRI KV Okha



Mr. Chirag Vankar, PMSHRI KV Junagadh



Mr. Mayank Gour, KV VV Nagar

*Turning
Ideas into
Learning
Resources*

*Organized
Relevant
Ready to Learn*

CONTENT COMPILATION TEAM



Ms. Huma Yasmin, K.V. No 4 ONGC, Vadodara



Ms. Neetika Bhardwaj, PMSHRI KV Porbandar



Mr. Parmendra Mangal, PMSHRI KV Dhrangadhra

*Compiling Knowledge...
Curating Excellence...*



IDEA



TEAMWORK



LEARNING



IMPACT

TOGETHER FOR A BRIGHTER TOMORROW

PYTHON REVISION TOUR

Section A — Multiple Choice Questions (14 × 1 mark)

- 1 What is the output of the functions shown below? 1
`min(max(False,-3,-4), 2,7)`
A. 2 B. False C. -3 D. -4
- 2 What will the following expression be evaluated to in Python? `float(5 + int(4.39 + 2.41) % 2)` 1
A. 5 B. 8.0 C. 5.0 D. 8
- 3 Consider the given expression: `not 5 or 4 and 10 and 'bye'` 1
Which of the following will be the correct output if the given expression is evaluated?
A. True B. False C. 10 D. 'bye'
- 4 Which operator has the highest precedence? 1
A. + B. * C. ** D. //
- 5 Which of the following is not a keyword in Python? 1
A. eval B. assert C. nonlocal D. pass
- 6 Identify the invalid Python statement from the following: 1
A. `d = dict()` B. `D = {}` C. `d = ()` D. `d = dict {}`
- 7 Which of the following expressions in Python evaluates to True? 1
A. `2>3 and 2<3` B. `3>1 and 2` C. `3>1 and 3>2` D. `3>1 and 3<2`
- 8 Which of the following statements is true about dictionaries in Python? 1
A. A dictionary is an example of sequence datatype.
B. A dictionary cannot have two elements with same key.
C. A dictionary cannot have two elements with same value.
D. The key and value of an element cannot be the same.
- 9 What is the value of the following expression ? 1
`3 + 3.00, 3**3.0`
A. `[6.0, 27.0]` B. `(6.0, 9.0)` C. `(6, 27)` D. `(6.0, 27.0)`
- 10 Which of the following is a keyword in Python ?
A. Break B. None C. For D. Class

Section B - Competency-Based Questions

- 11 What will be the value of **diff**: 1
- ```
c1='A'
c2='a'
diff= ord(c1)-ord(c2)
print(diff)
```
- A. Error: unsupported operator '-'      B. 32      C. -32      D. 0
- 12 What possible output(s) are expected to be displayed on screen at the time of execution of the program from the following code? 1
- ```
import random  
points=[20,40,10,30,15]  
points=[30,50,20,40,45]  
begin=random.randint(1,3)  
last=random.randint(2,4)  
for c in range(begin,last+1):  
    print(points[c],"#")
```
- A. 20#50#30# B. 20#40#45 C. 50#20#40# D. both (B. and (C.
- 13 What will be the output of the following python dictionary operation? 1
- ```
data = {'A':100, 'B':200, 'C':300, 'A':400}
print(data)
```
- A. {'A':100, 'B':200, 'C':300, 'A':400}      B. {'A':100, 'B':200, 'C':300}  
C. {'A':400, 'B':200, 'C':300}      D. It will generate an error
- 14 Select the correct output of the following python code: 1
- ```
str="My program is program for you"  
t = str.partition("program")  
print(t)
```
- A. ('My ', 'program', ' is ', 'program', ' for you')
B. ('My ', 'program', ' is program for you')
C. ('My ', ' is ', ' for you')
D. ('My ', ' is program for you')

- 15 Which of the following statement(s) would give an error during execution? 1
S=["CBSE"] # Statement 1
S+="Delhi" # Statement 2
S[0]= '@' # Statement 3
S=S+"Thank you" # Statement 4
- A. Statement 1 B. Statement 2 C. Statement 3 D. Statement 4
- 16 What output will the following program produce 1
v1='1'
v2= 1
V= v1==v2
print(V)
- A. Type Error B. Value Error C. True D. False
- 17 What will be the output of the following Python code? 1
try:
 num = 5
 text = "15"
 result = str(num) + text
 print(result)
except:
 print("Error")
- A. Error B. 515 C. 20 D. 75
- 18 What will be the output of the following statement? 1
print("X"* 2,"Y",end = 2 * "X", sep = "#")
- A. Error B. YY#XX C. XXYY# D. XX#YXX
- 19 Consider the statements given below and then choose the correct output from 1
the given options :
D={"A":"AHMEDABAD", "B":"BHAVNAGAR", "C":"CHENNAI"}
print("D" in D.
- A. True B. False C. 0 D. 1

- 20 Consider the statements given below and then choose the correct output from the given options: 1
- ```
K=('A','B')
V='TEST'
D = dict.fromkeys(K, V) print(D.
```
- A. {'A': 'TEST', 'B': None}                      B. {'A': None, 'B': 'TEST'}
- C. {'A': 'TEST', 'B': 'TEST'}                      D. Error
- 21 Amit wants to allow the program to repeatedly ask the user to enter their Choice if it does not equal the Answer. Which loop option should Amit use? 1
- A. while Choice = Answer:  
    Choice = input()
- B. while Choice != Answer:  
    Choice = input()
- C. while Choice != Answer:  
    Choice = input()
- D. while Choice == Answer:  
    Choice = input()
- 22 What will be the result of the following command: 1
- ```
>>>"i love python".capitalize()
```
- A. 'I Love Python' B. 'I love python'
- C. 'I LOVE PYTHON' D. None of the above
- 23 Which of the following statements will return the substring "STR" (i.e. the first 3 characters) from the string "STRING" ? 1
- A."STRING"[:-3] B."STRING"[-3::-1]
- C."STRING"[3:] D."STRING".startswith(3)
- 24 Which of the following statement(s) would give an error after executing the following code ? 1
- ```
Stud={"Murugan" : 100, "Mithu" : 95} # Statement 1 print
(Stud[95]) # Statement 2
Stud ["Murugan"]=99 # Statement 3
print(Stud.pop()) # Statement 4
print(StuD. # Statement 5
```
- A. Statement 2                      B. Statement 3
- C. Statement 4                      D. Statements 2 and 4

25 Given the following Tuple

1

Tup= (10, 20, 30, 50)

Which of the following statements will result in an error?

- A. print (Tup[0])
- B. Tup.insert (2, 3)
- C. print (Tup [1:2])
- D. print (len (Tup))

### Section C - Constructed-Response Questions

26 Julia has written a code to input a positive integer and display all its even factors in descending order. Her code is having errors. Rewrite the correct code and underline the corrections made.

2

```
n=input("Enter a positive integer: ")
for i in range(n):
 if i%2:
 if n%i==0:
```

```
 print(i, end=' ')
```

27 Predict the output of the following python code:

2

```
data = [2,4,2,1,2,1,3,3,4,4]
```

```
d = {}
```

```
for x in data:
```

```
 if x in d:
```

```
 d[x]=d[x]+1
```

```
 else:
```

```
 d[x]=1
```

```
print(D)
```

28 A. Given is a Python string declaration:

2

```
message='FirstCSWorkshop@2026-27'
```

Write the output of:

```
print(message[: -3].upper())
```

B. Write the output of the code given below:

```
d1={'rno':25, 'name':'dipanshu'}
```

```
d2={'name':'himanshu', 'age':30,'dept':'mechanical'}
```

```
d2.update(d1)
```

```
print(d2.keys())
```

29 What will be the output of the following python program?

2

```
str = ""
```

```
name = " New@Exam_26" for x
```

```
in name:
```

```
 if x in "aeiou": str+=x.upper()
```

```
 elif not x.isalnum():
```

```
 str+="**"
```

```
 elif x.isdigit():
```

```
 pass
```

```
 else:
```

```
 str+=x.lower()
```

```
print(str)
```

30 What possible output from the given options is expected to be displayed when the following code is executed ? Justify your answer. 2

2

```
import random
```

```
ITEMS={1:"PEN", 2: "ERASER", 3: "PENCIL"}
```

```
for i in range(len(ITEMS)): print(ITEMS[random.randint(0,i)+1],end="*")
```

A. PEN\*PENCIL\*PENCIL\*

B. B. PENCIL\*PEN\*ERASER\*

C. PEN\*PENCIL\*ERASER\*

D. PEN\*ERASER\*PEN\*

- 31 Differentiate between find() and index() with respect to strings in Python. 2  
Give supporting examples for both.
- 32 The code provided below is intended to input a positive integer from the user 2  
and display the total number of its factors. However, the code contains  
syntax and logical errors. Rewrite the code after removing all the errors.  
Underline all the corrections made.
- ```
n=int(input("Enter a positive integer:"))  
c=0  
for i in range(n+1):  
    if n%i=0:  
        c+=1  
print(C.
```
- 33 What possible output(s) are expected to be displayed on screen at the time 2
of execution of the following program :
- ```
import random
M=[5,10,15,20,25,30]
for i in range(1,3):
 first=random.randint(2,5)-1
 sec=random.randint(3,6)-2
 third=random.randint(1,4)
print(M[first], M[sec], M[third],sep="#")
```

- |             |              |
|-------------|--------------|
| A. 10#25#15 | B. 5#25#20   |
| 20#25#25    | 25#20#15     |
| C. 30#20#20 | D. 10#15#25# |
| 20#25#25    | 15#20#10#    |

34 What will be the output of following program:

2

```
s="welcome2kv"
n = len(s)
m=""
for i in range(0, n):
 if (s[i] >= 'a' and s[i] <= 'm'):
 m = m +s[i].upper()
 elif (s[i] >= 'n' and s[i] <= 'z'):
 m = m +s[i-1]
 elif (s[i].isupper()):
 m = m + s[i].lower()
 else:
 m = m +'#'
print(m)
```

35 (A) Given is a Python string declaration:

2

```
message='FirstPreBoardExam@2022-23'
```

Write the output of: `print(message[ : : -3].upper())`

(B) Write the output of the code given below:

```
d1={'rno':25, 'name':'dipanshu'}
d2={'name':'himanshu', 'age':30,'dept':'mechanical'}
d2.update(d1)
print(d2.keys())
```

36 Find and write the output of the following Python code :

2

```
Str1="PREBOARD2026"
Str2=""
I=0
while I<len(Str1):
 if Str1[I]>="A" and Str1[I]<="R":
 Str2=Str2+Str1[I+1]
 elif Str1[I]>="0" and Str1[I]<="9":
 Str2=Str2+ (Str1[I-1])
 else:
 Str2=Str2+"*"
I=I+1
print(Str2)
```

37 A. Predict the output of the code given below:

2

```
s="welcome2cs"
n = len(s)
m=""
for i in range(0, n):
 if (s[i] >= 'a' and s[i] <= 'm'):
 m = m +s[i].upper()
 elif (s[i] >= 'n' and s[i] <= 'z'):
 m = m +s[i-1]
 elif (s[i].isupper()):
 m = m + s[i].lower()
 else:
 m = m +'&'
print(m)
```

38 Sagar is writing a program to calculate the area of a triangle using heron's formula. He wrote his program like this -

4

```
a = input("Enter first side:")
b = input("Enter second side:")
c = input("Enter third side:")
area = sqrt(s(s-A)(s-B)(s-C))
print("Area of triangle = ",area)
```

(i) What is the required correction in the three input statements.

(ii) The expression -  $s = a+b+c / 2.0$  is syntactically correct but logically incorrect. Write the correct expression.

(iii) The program shows the error in the expression -

```
area = math.sqrt(s(s-A.(s-B.(s-C.).
```

Write the correct expression for it.

(iv) Write the module to be imported to successfully execute sqrt().

39 Raman is working on a python project – in one of the modules he has written the following code which is not executing properly 4

```
x = int(input("Enter the salary of July:"))
y = int(input("Enter the salary of December:"))
if y >= x:
 percent = (y-x) * 100 //x
 print("Salary Incremented by = ",y-x)
 print("Percentage of Salary incremented=",percent)
else:
 print("Salary not incremented.")
```

- (i) The interpreter shows the error at if statement. What is it ?
- (ii) When Raman has corrected the error at the if statement, the program shows another error – “Expected an indented block.” Why?
- (iii) The increment percentage in salary is not calculated correctly in the following run –

Enter the salary of July:12645

Enter the salary of December:13258

Salary Incremented by = 613

Percentage of Salary incremented= 4

Write the correct expression for percent variable. What would be the type of percent variable.

40 Predict the output on execution of the following Python code : 3

```
SCORES = [5, 12, 7, 20, 10, 18]
```

```
for X in range(len(SCORES)):
```

```
 if SCORES[X] % 4 == 0:
```

```
 SCORES[X] = SCORES[X] + SCORES[X-1]
```

```
 elif SCORES[X] % 3 == 0:
```

```
 SCORES[X] = SCORES[len(SCORES) - 1 - X] + 5
```

```
 else:
```

```
 SCORES[X] = SCORES[X] // 2
```

```
for Y in range(0, len(SCORES), 2):
 if Y + 1 < len(SCORES):
 print(SCORES[Y], SCORES[Y+1], sep="$")
 else:
 print(SCORES[Y], sep="$")
```

\*\*\*\*\*

## ANSWER KEY

### Select-Response Questions (MCQs)

- 1 **Answer:** B. False 1  
**EXPLANATION:** The function `max()` is being used to find the maximum value from among -3, -4 and false. Since false amounts to the value zero, hence we are left with `min(0, 2, 7)` Hence the output is 0 (false).
- 2 **Answer:** C. 5.0 1  
**EXPLANATION:**  $4.39 + 2.41 = 6.80$   
`int(6.80) = 6` (decimal part truncated).  
 $6 \% 2 = 0$   
 $5 + 0 = 5$   
`float(5) = 5.0`
- 3 **Answer:** D. 'bye' 1  
**Explanation:** Operator precedence: `not >` and `>` or. 'not 5' evaluates to False (since 5 is true). '4 and 10 and 'bye' evaluates left to right and, since all three are true, returns the LAST value, 'bye'. Finally, 'False or 'bye' returns 'bye' because the first operand is false.
- 4 **Answer:** C. \*\* 1  
**Explanation:** Among the given options, exponentiation (\*\*) has the highest precedence in Python's operator precedence table, evaluated before \*, //, and +.
- 5 **Answer:** A. eval 1  
**Explanation:** `eval` is a built-in FUNCTION used to evaluate a string as a Python expression; it is not a reserved keyword. `assert`, `nonlocal`, and `pass` are all genuine Python keywords.
- 6 **Answer:** D. `d = dict {}` 1  
**Explanation:** `d = dict()` and `D = {}` both correctly create an empty dictionary. `d = ()` is valid syntax (though it actually creates an empty TUPLE, not a dictionary — a semantic slip, not a syntax error). `d = dict {}` is a `SyntaxError`:  
'dict' is a built-in type/function and must be called with parentheses, `dict()`; writing a literal `{ }` directly after the name 'dict' without a call operator is invalid Python syntax.

7 **Answer:** C.  $3 > 1$  and  $3 > 2$

**Explanation:** A.  $2 > 3$  and  $2 < 3 \rightarrow$  False and True  $\rightarrow$  False.

(B.  $3 > 1$  and  $2 \rightarrow$  True and  $2 \rightarrow$  the expression's value is 2 (Python 'and' returns the actual operand, not necessarily a Boolean) — 2 is truthy, but it is not literally the value True.

(C.  $3 > 1$  and  $3 > 2 \rightarrow$  True and True  $\rightarrow$  True. (D.  $3 > 1$  and  $3 < 2 \rightarrow$  True and False  $\rightarrow$  False. Only option (C. evaluates to the literal Boolean True.

8 **Answer:** B. A dictionary cannot have two elements with the same key.

**Explanation:** A dictionary is a MAPPING type, not a sequence type (rules out A..

Keys in a dictionary must be unique — if a key is repeated during creation, the later value simply overwrites the earlier one (confirms B.. Values, however, are allowed to repeat (rules out C.. The key and the value of an element CAN be the same, e.g.  $\{ 'x': 'x' \}$  is perfectly valid (rules out D..

9 **Answer:** D. (6.0, 27.0)

**Explanation:** The comma makes this a tuple expression with two elements. First element:  $3 + 3.00 = 6.0$  (int promoted to float). Second element:  $3 * 3.0 = 27.0$  (exponentiation with a float exponent always yields a float). Hence the value is the tuple (6.0, 27.0).

10 **Answer:** B. None

**Explanation:** Python keywords are case-sensitive. 'Break', 'For', and 'Class' (with a capital first letter) are NOT keywords — the actual keywords are break, for, and class (all lowercase). None, however, IS a genuine keyword exactly as written (it represents Python's null value).

## Competency-Based Questions

11 **Answer:** C. -32

**Explanation:**  $\text{ord}('A') = 65$  and  $\text{ord}('a') = 97$  (uppercase letters have lower ASCII/Unicode values than their lowercase counterparts).  $\text{diff} = 65 - 97 = -32$ .

12 **Answer:** D. Both (B. and (C.

**Explanation:** The first 'points' list is immediately overwritten by the second assignment, so only `points = [30,50,20,40,45]` (indices 0-4) matters. Since the loop variable `c` runs from 'begin' to 'last' in INCREASING order, any valid output must be a contiguous, increasing-index slice of the values at indices 1-4, i.e.

a run selected from the sequence 50, 20, 40, 45 (index 0 = 30 is never reachable since begin >= 1).

Checking each option:

A. 20#50#30# is not a valid increasing-index run of this sequence and includes 30 (index 0), so it is impossible.

(B. 20#40#45# corresponds to begin=2, last=4 (indices 2,3,4) — a valid run. (C. 50#20#40# corresponds to begin=1, last=3 (indices 1,2,3) — also a valid run. Since both (B. and (C. are achievable depending on the random values chosen, the correct choice is (D..

13 **Answer:** C. {'A':400, 'B':200, 'C':300}

1

**Explanation:** When a dictionary literal contains a repeated key, Python keeps the key in its FIRST position but overwrites its value with the LAST assigned value for that key. Here 'A' appears twice; its final value is 400, and it retains its original (first) position in the dictionary.

14 **Answer:** B. ('My ', 'program', ' is program for you')

1

**Explanation:** The partition() method searches for only the FIRST occurrence of the given substring and splits the string into exactly a 3-element tuple: (text before the match, the matched substring itself, and everything after it) — it does NOT split on every occurrence, unlike split(). Here it splits only at the first 'program', giving ('My ', 'program', ' is program for you').

15 **Answer:** D. Statement 4

1

**Explanation:** Statement 2 (S += 'Delhi') works without error because += on a list accepts any ITERABLE, so it extends S with the individual characters 'D','e','l','h','i'. Statement 3 (S[0]='@') works because lists are mutable and item assignment by index is valid. Statement 4 (S = S + 'Thank you'), however, tries to concatenate a list with a string using '+', which Python does NOT allow (unlike +=) — this raises: TypeError: can only concatenate list (not "str") to list.

16 **Answer:** D. False

1

**Explanation:** v1 is the STRING '1' and v2 is the INTEGER 1. Python's == operator does not perform implicit type conversion between a string and a number for equality comparison — objects of different, unrelated types simply compare as not equal, so v1==v2 evaluates to False (no error is raised).

17 **Answer:** B. 515

1

**Explanation:** `str(5)` converts the integer 5 to the string "5". "5" + "15" performs string concatenation (not addition), producing "515". No exception occurs, so the `except` block is never reached.

18 **Answer:** D. XX#YXX

1

**Explanation:** "X"\*2 replicates to "XX", and 2\*"X" also replicates to "XX". The `print()` call therefore becomes `print("XX", "Y", end="XX", sep="#")`: the two values are joined using `sep="#"` to give "XX#Y", and then, instead of the usual newline, the custom `end="XX"` is appended, giving the final printed text "XX#YXX" (with no line break afterwards).

19 **Answer:** B. False

1

**Explanation:** The `in` operator checks **keys**, not values, when used directly with a dictionary.

20 **Answer:** C. {'A':'TEST', 'B':'TEST'}

1

**Explanation:** `dict.fromkeys(keys, value)` builds a new dictionary using every element of 'keys' as a key, with EVERY key mapped to the SAME single given value. Here, both 'A' and 'B' are assigned the identical value 'TEST'.

21 **Answer:** B. while Choice != Answer: Choice = input()

1

**Explanation:** The loop must keep executing WHILE Choice is NOT YET equal to Answer, so the condition needs the 'not equal to' operator, `!=`. Option A. uses `'='` (assignment), which is invalid inside a condition and causes a `SyntaxError`. Option (C. writes `'!='` , which is not a valid Python operator at all (also a `SyntaxError`). Option (D. uses `'=='`, which would run the loop only while the two ARE equal — the opposite of the required behaviour.

22 **Answer:** B. 'I love python'

1

**Explanation:** The `capitalize()` method capitalises ONLY the very first character of the entire string and forces every other character to lowercase — it does NOT capitalise the first letter of every word (that behaviour belongs to the `title()` method, which option A incorrectly describes).

23 **Answer:** A. "STRING"[: -3]

1

**Explanation:** "STRING" has 6 characters, so index -3 corresponds to positive index 3. The slice `[: -3]` therefore takes everything from the start up to (but excluding) index 3, giving "STR". Option B, `[-3::-1]`, starts at index 3 and steps

BACKWARDS, giving "IRTS". Option C, [3:], gives "ING" (the remaining half). Option D is invalid because startswith() expects a string argument, not an integer, and raises a TypeError.

24 **Answer:** D. Statements 2 and 4

**Explanation:** Statement 2, print(Stud[95]), raises a KeyError because 95 is not a KEY in the dictionary (the keys are the strings 'Murugan' and 'Mithu', not the number 95). Statement 3 is valid — it simply updates the existing key 'Murugan'. Statement 4, Stud.pop(), raises a TypeError because, unlike list.pop(), the dictionary pop() method REQUIRES a key argument and has no 'remove the last item' default behaviour.

25 **Answer:** B. Tup.insert(2, 3)

**Explanation:** Tuples are IMMUTABLE, so they support no methods that modify their contents — insert() is a list method and does not exist for tuples, raising AttributeError: 'tuple' object has no attribute 'insert'. Indexing (Tup[0]), slicing (Tup[1:2]), and len(Tup) are all read-only operations that work perfectly well on tuples.

### Constructed-Response Questions

26 Corrected code (corrections shown in the comments):

```
n = int(input("Enter a positive integer: ")) # int() added
for i in range(n, 0, -1): # descending, avoids i=0 if n
 % i == 0: # check factor first
 if i % 2 == 0: # fixed condition + indentation
 print(i, end=" ")
```

27 **Output:**

{2: 3, 4: 3, 1: 2, 3: 2}

**Explanation:** The code counts how many times each element occurs in the list 'data'. Element 2 occurs 3 times, 4 occurs 3 times, 1 occurs 2 times, and 3 occurs 2 times. The keys appear in the dictionary in the order they were FIRST encountered while scanning the list: 2 (index 0), then 4 (index 1), then 1 (index 3), then 3 (index 6).

28 A. **Output:** 762OKWTI

**Explanation:** message[::-3] starts from the LAST character and moves backwards, picking every 3rd character (step = -3). .upper() then converts

the resulting string to uppercase.

B. **Output:** dict\_keys(['name', 'age', 'dept', 'rno'])

**Explanation:** d2.update(d1) merges d1 into d2 — existing keys ('name') get their value overwritten by d1's value, and any new key from d1 not already in d2 ('rno') is appended at the end. The original relative order of d2's keys ('name','age','dept') is preserved, with the new key 'rno' added last.

29 **Output:** nEw\*\*exAm\*\*

**Explanation:** Each character of 'name' is processed in turn — vowels (a,e,i,o,u) are converted to UPPERCASE; any non-alphanumeric character (such as '@' or '\_') is replaced by "\*\*\*"; digits are skipped entirely (pass, so they contribute nothing to the result); and all remaining letters are converted to lowercase. Tracing through N-e-w-@-E-x-a-m-\_-2-6 gives: N-

>'n', e->'E' (vowel), w->'w', @->'\*\*\*', E->'E' (vowel), x->'x', a->'A' (vowel), m->'m', \_->'\*\*\*', 2 and 6 -> skipped. Concatenating gives nEw\*\*exAm\*\*.

30 **Answer:** (D. PEN\*ERASER\*PEN\*)

**Explanation:**

- Length of dictionary ITEMS is 3
- Loop runs 3 times with value of i as 0, 1, 2
- randint() will return 0 (for i=0) and 0 or 1 (for i=1) and 0 or 1 or 2 (for i=2)
- Key of ITEMS can be 1 (for i=0) and 1 or 2 (for i=1) and 1 or 2 or 3 (for i=2)
- First value can be PEN, second can be PEN or ERASER and third can be PEN or ERASER or PENCIL
- Hence only option (D. is a possible output of the given program.

31 **String.find(<Substring>)**

The find method searches for the first occurrence of the specified substring within the string. If found, it returns the index of the first occurrence the substring and in case the substring is not found, it returns - 1

Example :

T="Hello World"

print("'z' found at index:",T.find('z')) #

Output:'z' found at index: -1

**String.index(<Substring>)**

The index() method is similar to find(), but it behaves differently when the

substring is not found. If the substring is not found, it raises a ValueError.

Example :

```
T="Hello World"
```

```
print("'z' found at index:",T.index('z')) # will raise a ValueError
```

32 Corrected code:

```
n = int(input("Enter a positive integer: ")) # missing ")" added c = 0
for i in range(1, n+1): # start from 1, avoids n%0 error if n
 % i == 0: # fixed '=' to '=='
 c += 1 # fixed indentation
print(C.
```

33 **Answer:** (i) 10#25#15 and 20#25#25

**Justification:** random.randint(2,5)-1 always lies in {1,2,3,4};  
random.randint(3,6)-2 also always lies in {1,2,3,4};  
and random.randint(1,4) lies in {1,2,3,4} as well.

So EVERY value printed, on both lines (since the loop runs twice, printing once per iteration), must be M[1], M[2], M[3] or M[4] — i.e. one of 10, 15, 20, or 25. M[0]=5 and M[5]=30 can NEVER appear.

Option (ii) contains '5' and option (iii) contains '30', both of which are impossible, ruling them out. Option (iv) shows a trailing '#' after only three values (e.g. '10#15#25#'), which does not match a 3-argument print(..., sep="#") call (that produces exactly two '#' separators and no trailing one), so its formatting is inconsistent with the code. Only option (i), with both lines using values strictly from {10, 15, 20, 25} and correct formatting, is consistent with the code in every run.

34 **Output:** vELCcME#Kk

**Explanation:** For s = "welcome2kv", each character falling in the range 'a' to 'm' is upper-cased; each character in the range 'n' to 'z' is replaced by the PREVIOUS character in the original string (in lowercase, as it appears); the digit '2' matches none of the letter-range conditions and is also not uppercase, so it falls into the final else, contributing '#'. Tracing w-e-l-c-o-m-e-2-k-v gives: w(n-z)->s[-1]='v', e(a-m)->'E', l(a-m)->'L', c(a-m)->'C', o(n-z)->s[3]='c', m(a-m)->'M', e(a-m)->'E', 2->'#', k(a-m)->'K', v(n-z)-

>s[7]='2'... (using the ORIGINAL string s, not the growing s2, for the s[i-1] lookup)  
which, worked through carefully, yields vELCcME#Kk.

35 (A) 322ADORSF

(B) dict\_keys(['name', 'age', 'dept', 'rno'])

36 **Output:** REBOARD2D202

**Explanation:** For each character in 'PREBOARD2026', if it lies between 'A' and 'R' (an uppercase letter early in the alphabet), the NEXT character in Str1 is appended; if it is a digit ('0'-'9'), the PREVIOUS character is appended instead. Tracing through each character of P-R-E-B-O-A-R-D-2-0-2-6 and applying these rules in sequence builds up the string REBOARD2D202.

37 **Output:** sELCcME&Cc

**Explanation:** Identical logic to 34, applied to the string "welcome2cs" with '&' used in place of '#' for the non-alphabetic, non-uppercase character ('2'). Tracing through each character with the same rules (a-m -> uppercase itself; n-z -> previous character of the ORIGINAL string, lower-case; digit -> '&') produces sELCcME&Cc.

38 **(i) Correction required in the three input statements:**

Side lengths must be numeric to be used in arithmetic, but input() always returns a string. Each statement must convert the input to a float:

```
a = float(input("Enter first side:"))
```

```
b = float(input("Enter second side:"))
```

```
c = float(input("Enter third side:"))
```

**(ii) Correct expression for s (semi-perimeter):**

$s = a+b+c / 2.0$  is logically WRONG because of operator precedence: '/' binds more tightly than '+', so Python actually computes  $a + b + (c/2.0)$ , not  $(a+b+c)/2.0$ . The correct expression must use parentheses around the sum:

```
s = (a + b + c) / 2.0
```

(iii) Correct expression for the area:

`area = math.sqrt(s(s-A.(s-B.(s-C.)` is a `SyntaxError/TypeError` because writing `s(s-A.` attempts to CALL `s` as if it were a function; multiplication in Python must always use an explicit '\*' operator:

```
area = math.sqrt(s * (s-A. * (s-B. * (s-C.)
```

(iv) Module to be imported:

Since the corrected expression calls `srt()` as `math.srt()`, the `math` module must be imported at the top of the program: `import math`

### 39 (i) Error at the if statement:

'`y > = x`' has a stray space between '`>`' and '`=`', which Python does not recognise as a single operator; this causes a `SyntaxError`. It must be written as the single, unbroken comparison operator '`>=`': if `y >= x`:

**(ii) Why "Expected an indented block" occurs after fixing the if line:** Python defines a block's scope purely through consistent indentation. This error means that the line(s) immediately following the corrected '`if y >= x:`' are not indented at all (or are indented inconsistently), so the interpreter finds no statement belonging to the if block. All statements that are meant to execute when the condition is true must be indented uniformly (typically by 4 spaces) under the if line.

#### **(iii) Correct expression for percent:**

Using the floor-division operator '`//`' discards the decimal part of the result, which is why the sample run shows an incorrect, truncated value (4 instead of the true ~4.85%). Floor ('integer') division must be replaced with true (float) division, '`/`':

`percent = (y-x) * 100 / x`

Verification with the given sample values (`x=12645`, `y=13258`):  $(13258 - 12645) * 100 // 12645 = 4$  (incorrect, as printed.);  $(13258 - 12645) * 100 / 12645 = 4.847...$  (correct).

#### **(iv) Data type of percent:**

After the correction, since '`/`' (true division) always returns a floating-point result in Python regardless of the operand types, the variable `percent` will be of type `float`.

#### 40 Output:

2\$14

3\$23

5\$7

Explanation (step by step, list index by index):

- $X=0$ :  $\text{SCORES}[0]=5$ .  $5\%4!=0$ ,  $5\%3!=0$ , so else:  $\text{SCORES}[0]=5//2=2$ .

List: [2,12,7,20,10,18]

- $X=1$ :  $\text{SCORES}[1]=12$ .  $12\%4==0$ , so

$\text{SCORES}[1]=\text{SCORES}[1]+\text{SCORES}[0]=12+2=14$ .

List: [2,14,7,20,10,18]

- $X=2$ :  $\text{SCORES}[2]=7$ .  $7\%4!=0$ ,  $7\%3!=0$ , so else:  $\text{SCORES}[2]=7//2=3$ .

List: [2,14,3,20,10,18]

- $X=3$ :  $\text{SCORES}[3]=20$ .  $20\%4==0$ , so

$\text{SCORES}[3]=\text{SCORES}[3]+\text{SCORES}[2]=20+3=23$ .

List: [2,14,3,23,10,18]

- $X=4$ :  $\text{SCORES}[4]=10$ .  $10\%4!=0$ ,  $10\%3!=0$ , so else:  $\text{SCORES}[4]=10//2=5$ .

List: [2,14,3,23,5,18]

- $X=5$ :  $\text{SCORES}[5]=18$ .  $18\%4!=0$ ,  $18\%3==0$ , so

$\text{SCORES}[5]=\text{SCORES}[\text{len}(\text{SCORES})-1-5]+5=\text{SCORES}[0]+5=2+5=7$ . Final

list: [2,14,3,23,5,7]

- The second loop prints the final list two elements at a time: (2,14) -> '2\$14', (3,23) -> '3\$23', (5,7) -> '5\$7'.

**Mr. Santosh Kumar Deval, (PGT CS)**  
**PM SHRI KV Bhavnagar Para**

# WORKING WITH FUNCTIONS

## Section A — Multiple Choice Questions

1. Which keyword is used to define a function in Python?  
A. func  
B. def  
C. function  
D. lambda
2. What is the type of a built-in function like print, as returned by type(print)?  
A. <class 'str'>  
B. <class 'function'>  
C. <class 'builtin\_function\_or\_method'>  
D. Error
3. Which of the following is a Python built-in function?  
A. greeting()  
B. len()  
C. my\_func()  
D. calculate()
4. In the statement `def fun(x, y):`, x and y are called:  
A. Actual arguments  
B. Formal arguments / parameters  
C. Global variables  
D. Return values
5. Which type of argument allows passing multiple values using a single argument name?  
A. Default argument  
B. Keyword argument  
C. Variable-length argument  
D. Positional argument
6. What is the default return value of a function with no return statement?  
A. 0  
B. "" (empty string)  
C. None  
D. Error
7. Which keyword is used inside a function to modify a variable defined outside all functions?  
A. global  
B. nonlocal  
C. local  
D. public
8. In nested functions, which keyword allows a variable of the enclosing (outer) function to be modified?

A. global

B. nonlocal

C. outer

D. local

**9.** A variable declared inside a function is called a:

A. Global variable

B. Local variable

C. Static variable

D. Class variable

**10.** What is the output? ``def f(x): return x*2` `print(f(6))``

A. 6

B. 36

C. Error

D. 12

**11.** Which module must be imported to use mathematical functions like `srt()`?

A. statisticsma

B. th

C. random

D. numpy

**12.** What does `math.srt(4)` return?

A. 4

B. 2

C. 2.0

D. 4.0

**13.** Which of these is a mutable data type in Python?

A. tuple

B. string

C. list

D. int

**14.** Which of these is an immutable data type in Python?

A. list

B. dictionary

C. set

D. tuple

**15.** Given `def sum(x=3, y=4): return x+y` , what does `sum(x=4)` return?

A. 7

B. 8

C. 4

D. Error

**16.** A function that calls itself is called a:

A. Nested function

B. Recursive function

C. Lambda function

D. Anonymous function

**17.** What is the main advantage of using functions?

- A. They increase repetition                      B. They reduce reusability
- C. They promote code reuse and readability D. They eliminate variables

**18.** If an immutable object (e.g., an integer) is passed to a function and reassigned inside it, what happens to the original variable outside?

- A. It changes to the new value      B. It remains unchanged
- C. The program crashes      D. None of these

**19.** Which is the correct way to call a function using keyword arguments, for

```
def fun(name, age):?
```

- A. fun(15, "mohak")                      B. fun(age=15, name="mohak")  
C. fun[age=15]                              D. fun{age:15}

**20.** Which of the following best explains procedural abstraction as an advantage of functions?

- A. A programmer only needs to know how to call a function, not its internal code
- B. Programs run faster automatically
- C. Programs use less memory automatically
- D. None of these

## Section B – Fill in the Blanks

21. A function is defined in Python using the \_\_\_\_\_ keyword.

22. The \_\_\_\_\_ statement is used to send a result back from a function to the caller.

23. Variables declared inside a function have \_\_\_\_\_ scope.

24. \_\_\_\_\_ arguments assume a default value if no value is passed during the function call.

25. The \_\_\_\_\_ module needs to be imported to use functions like `srt()` and `factorial()`.

## Section C — True / False

- 26. Formal parameters are written in the function call.
- 27. Built-in functions do not need to be defined by the programmer.
- 28. Mutable objects, when passed to a function and modified inside it, reflect the change outside the function too.
- 29. The global keyword is used to access a variable of an immediately enclosing function (not the global scope).
- 30. A function in Python can return multiple values.

## Section D — Assertion & Reason Based Questions

Choose the correct option:

- (A) Both A and R are true, and R is the correct explanation of A.
  - (B) Both A and R are true, but R is not the correct explanation of A.
  - (C) A is true, but R is false.
  - (D) A is false, but R is true.
31. **Assertion (A):** Functions help increase code reusability.  
**Reason (R) :** A function can be called multiple times without rewriting its code.
32. **Assertion (A):** Local variables can be accessed outside the function in which they are declared.  
**Reason (R):** Local variables have scope limited to the function in which they are defined.
33. **Assertion (A):** The global keyword allows a function to modify a variable defined outside its scope.  
**Reason (R):** Without the global keyword, an assignment to a variable inside a function creates a new local variable.
34. **Assertion (A):** Lists are mutable, so when passed to a function and modified, the changes persist after the function call.  
**Reason (R):** Immutable objects always create a new object in memory when they are modified.

35. **Assertion (A):** A function with a return statement always returns a value to the caller.

**Reason (R):** If no return statement is present in a function, Python implicitly returns None.

36. **Assertion (A):** Keyword arguments allow passing arguments in any order in a function call.

**Reason (R):** The values in keyword arguments are matched to the parameter name, not to position.

37. **Assertion (A):** Variable-length arguments are useful when the number of arguments to be passed is not known in advance.

**Reason (R):** The \*args syntax collects extra positional arguments into a tuple.

38. **Assertion (A):** Every function in Python must have at least one argument.

**Reason (R):** Python allows functions to be defined without any parameters.

39. **Assertion (A):** The math module provides functions like `srt()`, `factorial()`, and `pow()`.

**Reason (R):** These functions are built into Python and do not require importing any module.

40. **Assertion (A):** A recursive function calls itself to solve a problem.

**Reason (R):** Every recursive function must have a base case to avoid infinite recursion.

## Answer Key — Chapter 2 (Functions)

| . No. | MC | . No. | Fill in Blank |
|-------|----|-------|---------------|
| 1     | B  | 21    | def           |
| 2     | C  | 22    | return        |
| 3     | B  | 23    | local         |
| 4     | B  | 24    | Default       |
| 5     | C  | 25    | math          |
| 6     | C  | 26    | False         |
| 7     | A  | 27    | True          |
| 8     | B  | 28    | True          |
| 9     | B  | 29    | False         |
| 10    | D  | 30    | True          |
| 11    | B  | 31    | A             |
| 12    | D  | 32    | D             |
| 13    | C  | 33    | A             |
| 14    | D  | 34    | B             |
| 15    | B  | 35    | C             |
| 16    | B  | 36    | A             |
| 17    | C  | 37    | A             |
| 18    | B  | 38    | D             |
| 19    | B  | 39    | C             |
| 20    | A  | 40    | A             |

**Mr. Bharat Katariya, (PGT CS)**  
**PMSHRI KV AFS Wadsar**

# EXCEPTION HANDLING

## Section A — Multiple Choice Questions

1. Errors that occur due to incorrect grammar/structure of code are called:  
A. Exceptions  
B. Syntax Errors  
C. Logical Errors  
D. Runtime Warnings
2. Which block is used to write code that might raise an exception?  
A. except  
B. try  
C. finally  
D. catch
3. Which exception is raised when a number is divided by zero?  
A. ValueError  
B. TypeError  
C. ZeroDivisionError  
D. NameError
4. Which exception occurs when a list index used is out of range?  
A. IndexError  
B. KeyError  
C. ValueError  
D. TypeError
5. Which exception is raised when a key is not found in a dictionary?  
A. IndexError  
B. KeyError  
C. NameError  
D. ValueError
6. Which block always executes, whether an exception occurs or not?  
A. try  
B. except  
C. else  
D. finally
7. Which exception is raised when a variable is used before it has been defined?  
A. NameError  
B. TypeError  
C. ValueError  
D. AttributeError
8. Which exception occurs when an operation is applied to an object of an incompatible type?  
A. ValueError  
B. TypeError  
C. KeyError  
D. ImportError

**9.** The else block in a try-except statement executes when:

- A. An exception occurs
- B. No exception occurs in the try block
- C. The finally block fails
- D. Always, before try

**10.** Which exception is raised when an import statement fails to find or load a module?

- A. ImportError
- B. SyntaxError
- C. NameError
- D. KeyError

**11.** Which exception is raised by the statement `int("abc")`?

- A. TypeError
- B. ValueError
- C. NameError
- D. KeyError

**12.** What is the correct order of blocks in Python exception handling?

- A. except, try, finally
- B. try, except, else, finally
- C. finally, try, except
- D. else, try, except

**13.** Which exception is raised when trying to access a non-existent attribute of an object?

- A. AttributeError
- B. ImportError
- C. IOError
- D. KeyError

**14.** Which type of error stops the program before it even starts executing?

- A. Exception
- B. Syntax Error
- C. Logical Error
- D. Runtime Error

**15.** Which exception is related to input/output operations, such as file access failures?

- A. IOError
- B. ValueError
- C. IndexError
- D. ImportError

**16.** Which statement about exceptions is correct?

- A. Exceptions always stop the program permanently
- B. Exceptions can be handled to keep the program running
- C. Exceptions only occur due to syntax errors
- D. Exceptions cannot be caught in Python

- 17.** Which of the following is a disadvantage of exception handling?
- A. Improved reliability
  - B. Simplified error handling
  - C. Performance overhead
  - D. Easier debugging
- 18.** Which keyword is used to catch a specific exception in Python?
- A. try
  - B. except
  - C. raise
  - D. finally
- 19.** For: `try: k = 9//0\nexcept ZeroDivisionError: print("Can't divide")\nfinally: print("Done")` — what is printed?
- A. "Can't divide" then "Done"
  - B. "Done" then "Can't divide"
  - C. Only "Done"
  - D. Only "Can't divide"
- 20.** Which statement best describes exception handling?
- A. A technique to prevent syntax errors
  - B. A mechanism to handle runtime errors gracefully without stopping the program
  - C. A way to make code run faster
  - D. A method used to declare variables

## Section B — Fill in the Blanks

- 21.** Errors that occur due to invalid code structure, such as a missing colon, are called \_\_\_\_\_ errors.
- 22.** The \_\_\_\_\_ block contains the code that might raise an exception.
- 23.** The \_\_\_\_\_ block always executes, irrespective of whether an exception occurred or not.
- 24.** \_\_\_\_\_ is the exception raised when division by zero is attempted.
- 25.** The \_\_\_\_\_ block executes only if no exception occurs in the try block.

## Section C — True / False

- 26. Syntax errors can be handled using try-except blocks.
- 27. The finally block executes only when an exception is raised.
- 28. ZeroDivisionError is raised when a number is divided by zero.
- 29. Exception handling improves the reliability of a program.
- 30. The except block is compulsory in every try statement.

## Section D — Assertion & Reason Based Questions

Choose the correct option:

- (A) Both A and R are true, and R is the correct explanation of A.
- (B) Both A and R are true, but R is not the correct explanation of A.
- (C) A is true, but R is false.
- (D) A is false, but R is true.

- 31. **Assertion (A):** Exceptions in Python can be handled using try-except blocks.  
**Reason (R):** Handling exceptions prevents abrupt termination of the program.
- 32. **Assertion (A):** Syntax errors can be caught and handled using except blocks.  
**Reason (R):** Syntax errors occur during the parsing of code, before execution begins.
- 33. **Assertion (A):** The finally block is always executed, regardless of whether an exception occurred.  
**Reason (R):** The finally block is mainly used to release resources, such as closing an open file.
- 34. **Assertion (A):** ZeroDivisionError occurs when a number is divided by zero.  
**Reason (R):** Division by zero is mathematically undefined, so Python raises an exception instead of returning an invalid result.
- 35. **Assertion (A):** The else block in a try-except statement is executed when an exception occurs.

**Reason (R):** The else block runs only if the try block does not raise any exception.

**36. Assertion (A):** Using exception handling always increases the speed of program execution.

**Reason (R):** Exception handling has some performance overhead compared to code without it.

**37. Assertion (A):** IndexError and KeyError are both related to accessing invalid positions in data structures.

**Reason (R):** IndexError is raised for sequences like lists, while KeyError is raised for dictionaries.

**38. Assertion (A):** A program with unhandled exceptions terminates abnormally.

**Reason (R):** Python does not provide any mechanism to catch runtime errors.

**39. Assertion (A):** Exception handling can make code longer and slightly more complex.

**Reason (R):** Exception handling can add performance overhead and coding complexity.

**40. Assertion (A):** An IOError may occur while reading from or writing to a file.

**Reason (R):** IOError is raised due to input/output failures, such as a missing file or a disk error.

## Answer Key — Chapter 3 (Exception Handling)

| . No. | MC | . No. | Fill in Blank     |
|-------|----|-------|-------------------|
| 1     | B  | 21    | Syntax            |
| 2     | B  | 22    | try               |
| 3     | C  | 23    | finally           |
| 4     | A  | 24    | ZeroDivisionError |
| 5     | B  | 25    | else              |
| 6     | D  | 26    | False             |
| 7     | A  | 27    | False             |
| 8     | B  | 28    | True              |
| 9     | B  | 29    | True              |
| 10    | A  | 30    | False             |
| 11    | B  | 31    | A                 |
| 12    | B  | 32    | D                 |
| 13    | A  | 33    | B                 |
| 14    | B  | 34    | A                 |
| 15    | A  | 35    | D                 |
| 16    | B  | 36    | D                 |
| 17    | C  | 37    | A                 |
| 18    | B  | 38    | C                 |
| 19    | A  | 39    | B                 |
| 20    | B  | 40    | A                 |

**Mr. Bharat Katariya, (PGT CS)**  
**PMSHRI KV AFS Wadsar**

# DATA FILE HANDLING (TEXT FILES)

## Section A — Multiple Choice Questions

1. Which mode is used to open a text file for reading only?  
A. w  
B. r  
C. a  
D. w+
2. What happens when a file is opened in w mode and the file already exists?  
A. New data is appended  
B. File is opened in read-only mode  
C. Existing contents are overwritten  
D. An error occurs
3. Which function reads the **entire contents** of a text file as a single string?  
A. readline()  
B. readlines()  
C. read()  
D. readall()
4. Consider:  

```
f = open("Data.txt", "r")
x = f.readline()
```

  
What does x contain?  
A. Entire file contents  
B. One line from the file  
C. A list of all lines  
D. Only the first character
5. Which function returns the contents of a text file as a **list of lines**?  
A. read()  
B. readline()  
C. readlines()  
D. getlines()
6. Which mode allows both **reading and writing** without automatically deleting the existing contents?  
A. r+  
B. w  
C. r  
D. w
7. Which mode is primarily used to add data at the **end** of an existing file?  
A. r  
B. w  
C. a  
D. r+
8. What is the main advantage of using the following statement?

with open("Test.txt", "r") as f:

data = f.read()

- A. It converts the file into binary format
- B. It automatically closes the file after the block
- C. It deletes the file after reading
- D. It opens the file twice

9. Which function is used to write a **single string** into a text file?

- A. write()                      B. writelines()                      C. writefile()                      D. put()

10. What does writelines() do?

- A. Writes multiple strings from an iterable
- B. Automatically adds a newline after every string
- C. Reads multiple lines
- D. Deletes existing lines

11. Suppose Data.txt contains:

Python

Computer

Science

What will be the output?

```
f = open("Data.txt", "r")
```

```
print(f.readline())
```

```
print(f.readline())
```

- A. Python Computer                      B. Python followed by Computer  
C. Python Computer Science                      D. Science

12. Which method returns the current position of the file pointer?

- A. position()                      B. pointer()                      C. tell()                      D. seek()

13. Which method is used to move the file pointer to a specified position?

- A. tell()                      B. seek()                      C. move()                      D. position()

14. Consider:

```
f = open("Data.txt", "r")
```

data = f.read(5)

What does data contain?

- A. First 5 lines      B. First 5 words      C. First 5 characters      D. Last 5 characters

### Section B — Fill in the Blanks

- 15.** The default mode of open() is \_\_\_\_\_ mode.
- 16.** The \_\_\_\_\_ method returns the current position of the file pointer.
- 17.** The \_\_\_\_\_ method is used to move the file pointer to a specified position.

### Section C — True / False

- 18.** readlines() returns a list containing the lines of a text file.
- 19.** Opening an existing file in w mode preserves its old contents.
- 20.** writelines() automatically adds \n after every string.

### Section D — Assertion & Reason Based Questions

Choose the correct option:

- (A) Both A and R are true, and R is the correct explanation of A.
- (B) Both A and R are true, but R is not the correct explanation of A.
- (C) A is true, but R is false.
- (D) A is false, but R is true.

- 21. Assertion (A):** Opening a file in w mode can result in loss of previously stored data.

**Reason (R):** w mode truncates an existing file before writing new data.

- 22. Assertion (A):** readline() reads one line at a time from a text file.

**Reason (R):** readline() returns the next line from the current file-pointer position.

- 23. Assertion (A):** tell() and seek() perform different operations on a file pointer.

**Reason (R):** tell() gives the current pointer position, whereas seek() changes the pointer position.

**24. Assertion (A):** The with statement is useful while working with files.

**Reason (R):** A file opened using with is automatically closed when the with block is exited.

**25. Assertion (A):** read() and readlines() always return the same type of value.

**Reason (R):** read() generally returns a string, whereas readlines() returns a list of lines.

### Answer Key

|    |                                              |    |                        |
|----|----------------------------------------------|----|------------------------|
| 1  | B — r                                        | 14 | C — First 5 characters |
| 2  | C — Existing contents are overwritten        | 15 | r                      |
| 3  | C — read()                                   | 16 | tell()                 |
| 4  | B — One line from the file                   | 17 | seek()                 |
| 5  | C — readlines()                              | 18 | True                   |
| 6  | A — r+                                       | 19 | False                  |
| 7  | C — a                                        | 20 | False                  |
| 8  | B — Automatically closes the file            | 21 | A                      |
| 9  | A — write()                                  | 22 | A                      |
| 10 | A — Writes multiple strings from an iterable | 23 | A                      |
| 11 | B — Python followed by Computer              | 24 | A                      |
| 12 | C — tell()                                   | 25 | D                      |
| 13 | B — seek()                                   |    |                        |

## TEXT FILE PROGRAMS

**26. Write a Python function COUNTCHAR() to read a text file STORY.TXT and count the total number of characters present in the file.**

```
def COUNTCHAR():
 f = open("STORY.TXT", "r")
 data = f.read()
 count = 0
 for ch in data:
 count += 1
 print("Number of characters:", count)
 f.close()
```

**27. Write a Python function COUNTWORDS() to read STORY.TXT and count the total number of words present in the file.**

```
def COUNTWORDS():
 f = open("STORY.TXT", "r")
 data = f.read()
 words = data.split()
 print("Number of words:", len(words))
 f.close()
```

**Important:** split() separates the text into words and returns a list.

Example: Python is very easy

split() → ['Python', 'is', 'very', 'easy']

So, number of words = 4.

**28. Write a Python function COUNTLINES() to read STORY.TXT and count the number of lines present in the file.**

```
def COUNTLINES():
 f = open("STORY.TXT", "r")
 lines = f.readlines()
 print("Number of lines:", len(lines))
 f.close()
```

**Alternative method:**

```
def COUNTLINES():
```

```
f = open("STORY.TXT", "r")
count = 0
for line in f:
 count += 1
print("Number of lines:", count)
f.close()
```

Both answers are valid.

**29. Write a Python function COUNTLINES() to count and display the number of lines in STORY.TXT that start with the letter 'A' or 'a'.**

```
def COUNTLINES():
 f = open("STORY.TXT", "r")
 count = 0
 for line in f:
 if line.startswith("A") or line.startswith("a"):
 count += 1
 print("Number of lines:", count)
 f.close()
```

**Alternate solution :** if line[0] == 'A' or line[0] == 'a':

**30. Write a Python function DISPLAYWORDS() to read STORY.TXT and display all words having fewer than 4 characters.**

```
def DISPLAYWORDS():
 f = open("STORY.TXT", "r")
 data = f.read()
 words = data.split()
 for word in words:
 if len(word) < 4:
 print(word)
 f.close()
```

**Example file:** This is my computer

**Output:**

is  
my

**31. Write a Python function COUNTWORD() to read STORY.TXT and count how many times the word "Python" occurs in the file.**

```
def COUNTWORD():
 f = open("STORY.TXT", "r")
 data = f.read()
 words = data.split()
 count = 0
 for word in words:
 if word == "Python":
 count += 1
 print("Python occurs", count, "times")
 f.close()
```

**If case should not matter:**

```
if word.lower() == "python":
```

**32. Write a Python function COUNT() to read STORY.TXT and count the number of vowels, consonants and spaces present in the file.**

```
def COUNT():
 f = open("STORY.TXT", "r")
 data = f.read()

 vowels = 0
 consonants = 0
 spaces = 0

 for ch in data:
 if ch.lower() in "aeiou":
 vowels += 1
 elif ch.isalpha():
 consonants += 1
 elif ch == " ":
 spaces += 1

 print("Vowels:", vowels)
 print("Consonants:", consonants)
 print("Spaces:", spaces)
 f.close()
```

**Most Important hints to Prepare**

| Program type                   | Main concept                |
|--------------------------------|-----------------------------|
| Count characters               | read() + loop               |
| Count words                    | read() + split()            |
| Count lines                    | readlines() / for line in f |
| Lines starting with A          | startswith() / line[0]      |
| Words < 4 characters           | split() + len()             |
| Count a particular word        | split() + comparison        |
| Count vowels/consonants/spaces | character-wise traversal    |

**Mr. A. S. Modi, (PGT CS)**  
**PMSHRI KV Sec.30 Gandhinagar**

# **DATA FILE HANDLING (BINARY & CSV FILES)**

## **Section A — Multiple Choice Questions**

1. Which module is commonly used in Python to work with binary files using object serialization?  
A. csv                                      B. pickle  
C. math                                      D. random
2. Which file mode opens a binary file for reading?  
A. r                                          B. rb  
C. w                                          D. wb
3. Which function is used to serialize a Python object into a binary file?  
A. pickle.dump()                      B. pickle.load()  
C. csv.writer()                      D. csv.reader()
4. Which function reads an object from a binary file created using pickle?  
A. pickle.read()                      B. pickle.load()  
C. pickle.input()                      D. pickle.get()
5. Which module is used for reading and writing CSV files in Python?  
A. pickle                                  B. csv  
C. os                                          D. sys
6. Which function creates a writer object for writing rows to a CSV file?  
A. csv.reader()                      B. csv.open()  
C. csv.writer()                      D. csv.write()
7. Which method is generally used to write a list of values as one row in a CSV file?  
A. writerow()                          B. writecolumn()  
C. addrow()                              D. insertrow()

## Section B – Assertion and Reasoning

Choose the correct option:

- (A) Both A and R are true, and R is the correct explanation of A.
- (B) Both A and R are true, but R is not the correct explanation of A.
- (C) A is true, but R is false.
- (D) A is false, but R is true.

8. Assertion (A): A binary file can store Python objects in a form that can be retrieved later using pickle.

Reason (R): `pickle.dump()` serializes an object and writes it to a binary file.

9. Assertion (A): The `csv.reader()` function is used to write rows into a CSV file.

Reason (R): `csv.writer()` is used to write data into a CSV file.

10. Assertion (A): CSV files store data in rows and columns, with fields commonly separated by a delimiter such as a comma.

Reason (R): `csv.reader()` can read a CSV file row by row.

## Section C – Fill in the Blanks

- 11. The Python module used for binary file serialization is \_\_\_\_\_.
- 12. The file mode used to open a binary file for writing is \_\_\_\_\_.
- 13. The \_\_\_\_\_ function retrieves an object from a pickle file.
- 14. The Python module used to handle CSV files is \_\_\_\_\_.
- 15. The \_\_\_\_\_ method writes a single row to a CSV file using a writer object.

## Section D – True / False

- 16. `pickle.dump()` is used to write/serialize a Python object into a binary file.
- 17. A binary file must always be opened using text mode 'r' or 'w'.
- 18. `csv.reader()` is used to read rows from a CSV file.
- 19. `csv.writer()` can be used to write rows into a CSV file.
- 20. CSV stands for Comma Separated Values.

## ANSWER KEY

|    |   |    |               |
|----|---|----|---------------|
| 1  | B | 11 | pickle        |
| 2  | A | 12 | wb            |
| 3  | B | 13 | pickle.load() |
| 4  | B | 14 | csv           |
| 5  | B | 15 | writerow()    |
| 6  | C | 16 | True          |
| 7  | A | 17 | False         |
| 8  | A | 18 | True          |
| 9  | D | 19 | True          |
| 10 | B | 20 | True          |

# **DATA FILE HANDLING (Binary File )**

## **Section A – 2 Marks questions**

- 21.** Differentiate between a text file and a binary file. Also write the Python file modes used to open a binary file for reading and writing.
- 22.** What is the purpose of the pickle module in Python? Explain the use of pickle.dump() and pickle.load() with suitable syntax.
- 23.** Consider the following code:
- ```
import pickle
f = open("student.dat", "wb")
rec = [101, "Riya", 92]
pickle.dump(rec, f)
f.close()
```

Answer the following:

- A. What type of file is student.dat?
- B. What will pickle.dump() store in the file?
- 24.** Write a Python statement to open a binary file named "Employee.dat" in:
(A) binary read mode
(B) binary append mode
Also state why the append mode is useful in a binary file.
- 25.** Why is EOFError commonly handled while reading records from a binary file using pickle.load()? Write the relevant exception name and the purpose of handling it.

Section B – 3 Marks Questions

- 26.** A binary file "STUDENT.DAT" contains records in the form [RollNo, Name, Marks]. Write a Python function SearchStudent() that accepts a roll number and displays the corresponding Name and Marks. If the roll number is not found, display an appropriate message.
- 27.** Write a Python function AddStudent() to append records to a binary file "STUDENT.DAT". Each record should contain RollNo, Name and Marks.
- 28.** A binary file "STUDENT.DAT" contains records [RollNo, Name, Marks]. Write a Python function UpdateMarks() that searches for a given RollNo and updates its Marks. Assume that the file contains a list of records and use an appropriate approach for updating the binary file.

29. Study the following code and answer the questions:

```
import pickle
with open("data.dat", "rb") as f:
    while True:
        try:
            rec = pickle.load(f)
            print(reC)
        except EOFError:
            break
```

- (A) Why is the file opened in "rb" mode?
- (B) What does pickle.load(f) do?
- (C) Why is EOFError handled?

30. A school maintains a binary file "RESULT.DAT" containing records [RollNo, Name, Marks]. Write a function CountHigh() to count and display the number of students whose marks are greater than or equal to 75. The function should read all records using pickle.load().

Answers – Section A (2 Mark Questions)

21. A text file stores data as readable characters, whereas a binary file stores data in binary/byte form. Binary files are commonly opened using rb for reading and wb for writing.

22. The pickle module is used for serialization and deserialization of Python objects. pickle.dump(object, file) serializes and stores an object; pickle.load(file) reads and reconstructs the stored object.

23. (A) student.dat is a binary file. (B) The list [101, 'Riya', 92] is serialized and stored as a Python object in binary form.

24. (A) f = open("Employee.dat", "rb") (B) f = open("Employee.dat", "ab").

Append mode adds new records at the end of the existing binary file without removing existing records.

25. pickle.load() keeps reading objects until the end of the file is reached. At the end, EOFError is raised. Handling EOFError allows the program to stop reading normally instead of terminating with an error.

Answers – Section B (3 Marks Questions)

26.

```
import pickle
```

```
def SearchStudent():
    r = int(input("Enter Roll No: "))
    found = False
    with open("STUDENT.DAT", "rb") as f:
        try:
            while True:
                rec = pickle.load(f)
                if rec[0] == r:
                    print("Name:", rec[1])
                    print("Marks:", rec[2])
                    found = True
                    break
        except EOFError:
            pass

    if not found:
        print("Record not found")
```

27.

```
import pickle
```

```
def AddStudent():
    with open("STUDENT.DAT", "ab") as f:
        r = int(input("Enter Roll No: "))
        name = input("Enter Name: ")
        marks = float(input("Enter Marks: "))
        rec = [r, name, marks]
        pickle.dump(rec, f)
```

28.

```
import pickle
def UpdateMarks():
    r = int(input("Enter Roll No: "))
    new_marks = float(input("Enter New Marks: "))
    records = []
    found = False
```

```

with open("STUDENT.DAT", "rb") as f:
    try:
        while True:
            rec = pickle.load(f)
            if rec[0] == r:
                rec[2] = new_marks
                found = True
                records.append(rec)
    except EOFError:
        pass

with open("STUDENT.DAT", "wb") as f:
    for rec in records:
        pickle.dump(rec, f)

if found:
    print("Marks updated successfully")
else:
    print("Record not found")

```

29. (A. rb means read in binary mode.

(B. pickle.load(f) reads and reconstructs one serialized Python object from the binary file.

(C. EOFError is raised when there are no more objects to read, so it is handled to terminate the loop normally.

30.

```

import pickle

def CountHigh():
    count = 0
    with open("RESULT.DAT", "rb") as f:
        try:
            while True:
                rec = pickle.load(f)
                if rec[2] >= 75:
                    count += 1
        except EOFError:
            pass

    print("Number of students:", count)

```

Ms. Renu Baheti, (PGT CS)
PMSHRI KV Okha

DATA FILE HANDLING (CSV File)

Section A – 4 Marks Questions

31. Sangeeta is a Python programmer working in a computer hardware company. She has to maintain the records of the peripheral devices. She created a csv file named Peripheral.csv, to store the details. The structure of Peripheral.csv is [P_id, P_name, Price] where P_id is Peripheral device ID (integer), P_name is Peripheral device name (String), Price is Peripheral device price (integer). Sangeeta wants to write the following user defined functions: Add_Device(): to accept a record from the user and add it to a csv file, Peripheral.csv. Count_Device(): To count and display number of peripheral devices whose price is less than 1000.

Answer:

```
import csv
def Add_Device():
    F = open("Peripheral.csv", "a", newline="")
    W = csv.writer(F)
    P_id = int(input("Enter the Peripheral ID"))
    P_name = input("Enter Peripheral Name")
    Price = int(input("Enter Price"))
    L = [P_id, P_name, Price]
    W.writerow(L)
    F.close()
def Count_Device():
    F = open("Peripheral.csv", "r")
    L = list(csv.reader(F))
    Count = 0
    for D in L:
        if int(D[2]) < 1000:
            Count += 1
    print(Count)
    F.close()
```

32. Write a Program in Python that defines and calls the following user defined functions: (i) add() – To accept and add data of a Product to a CSV file, Product.csv'. Each record consists of a list with field elements as pid, pname and price to store Product id, Product name and Product price respectively. (ii) search() – To display the records of the Product whose price is more than 20000 and product name starting with a vowel.

Answer:

```
import csv
def add():
    l = []
    for i in range(4):
        pid = input("Enter pid")
        pname = input("Enter Pname")
        price = int(input("Enter price of product"))
        p = [pid, pname, price]
        l.append(p)
    f = open("product.csv", 'w', newline='')
    w = csv.writer(f)
    w.writerows(l)
    f.close()
def search():
    f = open("product.csv", 'r')
    r = csv.reader(f)
    for i in r:
        if int(i[2]) > 20000 and i[1][0] in "AEIOUaeiou":
            print(*i)
    f.close()
add()
search()
```

- 33.** A csv file "Employment.csv" contains the data of a survey. Each record of the file contains: Name of a state, Population of the state, Sample size (number of persons who participated in the survey in that state), Employed (number of persons employed). For example, a sample record of the file may be: ['Rajasthan', 5674933, 10000, 5679]. Write the following Python functions to perform the specified operations on this file: (i) Read all the data from the file in the form of a list and display all those records for which the population is more than 5000000. (ii) Count the number of records in the file.

Answer:

```
import csv
def show():
    f = open("Employment.csv")
    ro = csv.reader(f)
    for rec in ro:
        if int(rec[1]) > 5000000:
            print(rec)
    f.close()
```

```
def count_records():
    f = open("Employment.csv")
    ro = csv.reader(f)
    l = list(ro)
    print("no. of records = ", len(l))
    f.close()
```

34. A librarian is managing book inventory using a CSV file named Inventory.csv. The file structure is [BookID, Title, Author, Available] where BookID is an integer, Title and Author are strings, and Available is an integer representing the number of copies available. The librarian needs to write the following functions:
 add_book(): accepts new book details from the user and adds them to Inventory.csv — the file should be created with column headers if it doesn't exist.
 check_availability(book_id.): takes a book_id as input and returns the number of copies available for that book; if the book is not found, it should return -1.

Answer:

```
import csv
import os
def add_book():
    file_exists = os.path.isfile('Inventory.csv')
    with open('Inventory.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        if not file_exists:
            writer.writerow(['BookID', 'Title', 'Author', 'Available'])
        book_id = input("Enter BookID: ")
        title = input("Enter Title: ")
        author = input("Enter Author: ")
        available = input("Enter number of copies available: ")
        writer.writerow([book_id, title, author, available])
    print("Book added successfully!")

def check_availability(book_id.):
    try:
        with open('Inventory.csv', 'r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['BookID'] == str(book_id.):
                    return int(row['Available'])
    except FileNotFoundError:
        print("Inventory file not found.")
    return -1
```

35. Write the following Python functions to perform the specified operations on a CSV file of players: (i) Read all the data from the file in the form of a list and display records of all the players, whose country is 'India'. (ii) Count the number of records in the file.

Answer:

```
import csv
```

```
def show():
```

```
    f = open("players.csv", 'r')
    records = csv.reader(f)
    next(records, None)      # To skip the Header row
    for i in records:
        if i[3] == 'India':
            print(i)
    f.close()
```

```
def Count_records():
```

```
    f = open("players.csv", 'r')
    records = csv.reader(f)
    next(records, None)      # To skip the Header row
    count = 0
    for i in records:
        count += 1
    print(count)
    f.close()
```

36. Search an employee by name from CSV file. Each record consists of a list with field elements as empid, name, salary and dname to store employee id, employee name, employee salary and employee department name respectively.

Answer:

```
import csv
```

```
def search():
```

```
    f=open("xyz.csv",'r')
    data= csv.reader (f)
    n=input("enter name to be searched")
    for i in data:
        if i[1]==n:
            print(i)
    f.close()
```

37. Display and count the employees whose salary is more than 50000 from CSV file. Each record consists of a list with field elements as empid, name , salary and dname to store employee id, employee name, employee salary and employee department name respectively.

Answer:

```
import csv
def search():
    f=open("xyz.csv",'r')
    data= csv.reader (f)
    count=0
    for i in data:
        if i[2]>50000:
            print(i)
            count=count+1
    print ("number of employees getting more than 50000=",count)
    f.close()
```

38. Accept department name as parameter in search function and display and count employees of that department from CSV file. Each record consists of a list with field elements as empid, name , salary and dname to store employee id, employee name, employee salary and employee department name respectively.

Answer:

```
import csv
def search(dept):
    f=open("xyz.csv",'r')
    data= csv.reader (f)
    count=0
    for i in data:
        if i[3]== dept:
            print(i)
            count=count+1
    print ("number of employees",count)
    f.close()

search("sales")
```

39. Mr. Mahesh is a Python Programmer working in a school. He has to maintain the records of the sports students. He has created a csv file named sports.csv, to store the details. The structure of sports.csv is :
[sport_id, competition, prize_won] where sport_id, is Sport id (integer)
competition is competition name (string)
prize_won is ("Gold", "Silver", "Bronze")
Mr. Mahesh wants to write the following user-defined functions :
Add_detail(): to accept the detail of a student and add to a csv file,"sports.csv".
Count_Medal(): to display the name of competitions in which students have won "Gold" medal.
Help him in writing the code of both the functions.

Answer:

```
import csv
def Add_detail():
    F=open("sports.csv","a")
    W=csv.writer(F)
    sport_id=int(input("Sport id:"))
    competition=input("Competition:")
    prize_won=input("Prize won:")
    L=[sport_id,competition,prize_won]
    W.writerow(L)
    F.close( )
def Count_Medal( ):
    F=open("sports.csv","r")
    L=list(csv.reader(F))
    for D in L:
        if D[2]=="Gold":
            print("Competition:",D[1])
    F.close()
```

40. Write a program in Python that defines and calls the following user defined functions:
- (i) COURIER_ADD() : It takes the values from the user and adds the details to a csv file 'courier.csv'. Each record consists of a list with field elements as cid, s_name, Source, destination to store Courier ID, Sender name, Source and destination address respectively.
 - (ii) COURIER_SEARCH() : Takes the destination as the input and displays all the courier records going to that destination

Answer:

```
import csv
def COURIER_ADD() :
    f1=open("courier.csv","a",newline="\n")
    writ=csv.writer(f1)
    cid=int(input("Enter the Courier id"))
    s_name=input ("Enter the Sender Name")
    Source=input("Enter the Source Address")
    destination=input("Enter Destination Name")
    detail=[cid, s_name, Source, destination]
    writ.writerow (detail)
    f1.close()
def COURIER_SEARCH():
    f1=open("courier.csv", "r") # ignore newline
    detail=csv.reader(f1)
    name=input("Enter the Destination Name to searched")
    for i in detail :
        if i[3]==name:
            print("Details of courier are: ",i)
    f1.close()
```

Ms. Renu Baheti, (PGT CS)
PMSHRI KV Okha

STACK

Section A — Multiple Choice questions

1. A stack follows which principle?
A. FIFO B. LIFO C. FILO only D. Random access
2. Which operation is used to **insert** an element into a stack?
A. POP B. PUSH C. INSERT D. ADD
3. Which operation removes an element from a stack?
A. PUSH B. INSERT C. POP D. DELETE
4. Consider the following Python code:

```
S = []  
S.append(10)  
S.append(20)  
S.append(30)  
print(S.pop())
```

What will be the output?

- A. 10 B. 20 C. 30 D. [10, 20]
5. Which Python list method is generally used to implement **PUSH** operation?
A. pop() B. append() C. insert() only D. remove()
 6. Which Python list method is generally used to implement **POP** operation?
A. delete() B. remove() C. pop() D. discard()
 7. Consider:

```
S = [5, 10, 15]  
S.pop()  
S.append(20)  
print(S)
```

What will be the output?

- A. [5, 10, 15, 20]
B. [5, 10, 20]
C. [10, 15, 20]
D. [5, 20]

8. What happens when pop() is performed on an **empty stack** implemented using a Python list?

- A. Underflow condition occurs
- B. Overflow occurs
- C. 0 is returned
- D. None is automatically inserted

9. The insertion operation in a stack is called _____.

10. The deletion operation in a stack is called _____.

11. In a stack, the element inserted last is removed first. **True / False**

12.

Assertion (A): A stack is called a LIFO data structure.

Reason (R) : The element inserted last into the stack is the first element to be removed.

Choose:

- A. Both A and R are true, and R is the correct explanation of A.
- B. Both A and R are true, but R is not the correct explanation of A.
- C. A is true, but R is false.
- D. A is false, but R is true.

Answer Key

1	B — LIFO	7	B — [5, 10, 20]
2	B — PUSH	8	A — Underflow condition
3	C — POP	9	PUSH
4	C — 30	10	POP
5	B — append()	11	True
6	C — pop()	12	A

STACK – PROGRAMS

13. Write a Python program to implement a stack using a list. Define functions

PUSH() and POP().

```
Stack = []
def PUSH():
    x = int(input("Enter number: "))
    Stack.append(x)

def POP():
    if Stack == []:
        print("Stack Underflow")
    else:
        print("Deleted element:", Stack.pop())
```

```
PUSH()
PUSH()
POP()
print("Stack:", Stack)
```

Important points:

append() → PUSH

pop() → POP

Empty stack → **Underflow**

14. Menu-Driven Stack

Write a Python program to implement a **menu-driven stack** with the following operations:

1. PUSH
2. POP
3. DISPLAY
4. EXIT

```
Stack = []
def PUSH():
    x = int(input("Enter number: "))
    Stack.append(x)

def POP():
    if len(Stack) == 0:
        print("Stack Underflow")
    else:
        print("Deleted:", Stack.pop())
```

```

def DISPLAY():
    if len(Stack) == 0:
        print("Stack is empty")
    else:
        print("Stack:", Stack)

while True:
    print("\n1. PUSH")
    print("2. POP")
    print("3. DISPLAY")
    print("4. EXIT")

    ch = int(input("Enter choice: "))
    if ch == 1:
        PUSH()
    elif ch == 2:
        POP()
    elif ch == 3:
        DISPLAY()
    elif ch == 4:
        break
    else:
        print("Invalid choice")

```

15 Stack of Even Numbers

Write a function PUSH() to accept a number and push it into the stack **only if it is even**. Also write POP() to remove the top element.

```

Stack = []
def PUSH():
    n = int(input("Enter number: "))

    if n % 2 == 0:
        Stack.append(n)
    else:
        print("Only even numbers can be pushed")

def POP():
    if Stack == []:
        print("Stack Underflow")
    else:
        print("Popped element:", Stack.pop())

PUSH()
PUSH()
PUSH()
print("Stack:", Stack)

```

```
POP()
print("Stack:", Stack)
```

16. Stack of Words — Push Words Starting with 'A'

Write a Python function PUSH() that accepts a word and pushes it into a stack **only if it starts with the letter 'A' or 'a'**. Also define POP() to remove and display the top word.

```
Stack = []
def PUSH():
    word = input("Enter word: ")

    if word[0] == 'A' or word[0] == 'a':
        Stack.append(word)
        print("Word pushed")
    else:
        print("Word not pushed")

def POP():
    if len(Stack) == 0:
        print("Stack Underflow")
    else:
        print("Popped word:", Stack.pop())
```

```
PUSH()
PUSH()
PUSH()
POP()
```

Example

Input:

Apple

Ball

Ant

Stack:

Apple

Ant

17. Find and Display the Largest Element in a Stack

Write a Python program to create a stack of numbers. Define a function LARGEST() to find and display the **largest element** in the stack.

```
Stack = []
def PUSH():
    n = int(input("Enter number: "))
    Stack.append(n)
```

```
def LARGEST():
    if len(Stack) == 0:
        print("Stack is empty")
    else:
        largest = Stack[0]

        for i in Stack:
            if i > largest:
                largest = i

        print("Largest element:", largest)
```

```
PUSH()
PUSH()
PUSH()
PUSH()
print("Stack:", Stack)
LARGEST()
```

18. Consider a list named Nums which contains random integers. Write the following user defined functions in Python and perform the specified operations on a stack named BigNums.

- (i) PushBig(): It checks every number from the list Nums and pushes all such numbers which have 5 or more digits into the stack, BigNums.
- (ii) PopBig(): It pops the numbers from the stack, BigNums and displays them. The function should also display "Stack Empty" when there are no more numbers left in the stack.

For example: If the list Nums contains the following data:

Nums = [213, 10025, 167, 254923, 14, 1297653, 31498, 386, 92765]

Then on execution of PushBig(), the stack BigNums should store:

[10025, 254923, 1297653, 31498, 92765]

And on execution of PopBig(), the following output should be displayed:

92765

31498

1297653

254923

10025

Stack Empty

Answer :

```
def PushBig(Nums,BigNums):  
    for N in Nums:  
        if len(str(N)) >= 5:  
            BigNums.append(N)
```

```
def PopBig(BigNums):  
    while BigNums:  
        print(BigNums.pop( ))  
    else:  
        print("Stack Empty")
```

19. A list contains following record of course details for a University:

[Course_name, Fees, Duration]

Write the following user defined functions to perform given operations on the stack named 'Univ' :

- (i) Push_element() - To push an object containing the Course_name, Fees and Duration of a course, which has fees greater than 100000 to the stack.
- (ii) Pop_element() - To pop the object from the stack and display it. Also, display "Underflow" when there is no element in the stack.

For example:

If the lists of courses details are:

["MCA", 200000, 3]

["MBA", 500000, 2]

["BA", 100000, 3]

The stack should contain

["MBA", 500000, 2]

["MCA", 200000, 3]

Answer :-

```
Univ=[]  
def Push_element(Course):  
    for Rec in Course:  
        if Rec[1]>100000:  
            Univ.append(Rec)
```

```
def Pop_element( ):
    while len(Univ)>0:
        print(Univ.pop( ))
    else:
        print("Underflow")
```

20. Write separate user defined functions for the following:

- (i) PUSH(N) - This function accepts a list of names, N as parameter. It then pushes only those names in the stack named OnlyA which contain the letter 'A'.
- (ii) POPA(OnlyA) - This function pops each name from the stack OnlyA and displays it. When the stack is empty, the message "EMPTY" is displayed.

For example :

If the names in the list N are

['ANKITA', 'NITISH', 'ANWAR', 'DIMPLE', 'HARKIRAT']

Then the stack OnlyA should store

['ANKITA', 'ANWAR', 'HARKIRAT']

And the output should be displayed as

HARKIRAT ANWAR ANKITA EMPTY

Answer :-

OnlyA=[]

def PUSH(N):

 for aName in N :

 if 'A' in aName :

 OnlyA.append(aName)

def POPA(OnlyA):

 while OnlyA :

 print(OnlyA.pop(), end=' ')

 else :

 print('EMPTY')

Mr. A. S. Modi, (PGT CS)
PMSHRI KV Sec.30 Gandhinagar

COMPUTER NETWORKS

Section A — Multiple Choice questions

1. Which one of the below is a protocol used for retrieving emails from a mail server.
A. SMTP B. FTP C. POP3 D. PPP
2. Which of the following is correct about using a Hub and Switch in a computer network?
A. A hub sends data to all devices in a network, while a switch sends data to the specific device.
B. A hub sends data only to the devices it is connected to, while a switch sends data to all devices in a network.
C. A hub and switch function the same way and can be used interchangeably.
D. A hub and switch are both wireless networking devices.
3. Which of the following is used to create the structure of a web page?
A. CSS B. HTML C. JavaScript D. FTP
4. Which device is primarily used to amplify and regenerate signals in a network, allowing data to travel longer distances?
A. Switch B. Router C. Repeater D. Bridge
5. Ethernet card is also known as:
A. LIC B. MAC C. NIC D. OIC
6. Which protocol is used by mail servers and email clients to send email messages across the internet?
A. HTTP B. FTP C. POP3 D. SMTP
7. Which of the following devices is essential to set up a wired LAN?
A. Modem B. NIC C. Repeater D. Firewall
8. _____ is used for point-to-point communication or unicast communication such as radar and satellite.
A. Bluetooth B. Infrared C. Radio waves D. Microwaves
9. Which network device serves as the entry and exit point of a network, as all data coming in or going out of a network must first pass through it?
A. Gateway B. Modem C. Switch D. Repeater

10. Which of the following Networking Protocol is used for remote login.
A. SMTP B. FTP C. POP3 D. TELNET
11. Which of the following statements is correct about using a Router and a Gateway in a computer network?
A. A router connects multiple networks and directs data between them, while a gateway connects networks using different protocols.
B. A router and a gateway perform the same function and can be used interchangeably.
C. A router connects devices within same network, while a gateway connects wireless networks.
D. A router connects two networks while gateway disconnects different networks
12. Which protocol facilitates remote login to any PC?
A. SMTP B. FTP C. VOIP D. TELNET
13. What is the use of Bridge in Network?
A. to connect LANs B. to separate LANs
C. to control Network Speed D. All of the above
14. Which network topology requires a central controller or hub?
A. Star B. Mesh C. Ring D. Bus
15. Sunil has purchased a new Smart TV and wants to cast a video from his mobile to his new Smart TV. Identify the type of network he is using.
A. MAN B. LAN C. PAN D. None of the above
16. Which of the following options is the correct protocol used for phone calls over the Internet?
A. PPP B. FTP C. HTTP D. VoIP
17. Which of the following is the smallest network?
A. WAN B. MAN C. LAN D. PAN
18. What does HTTPs stand for?
A. Hyper Text Protocol Secure B. Hypertext Transfer Protocol Secure
C. Hidden Text Transfer Protocol Secure D. None of the above
19. Which of these is not an example of Unguided Media?
A. Optical Fibre Cable B. Radio Waves
C. Infrared D. Satellite

20. One Device per connection is an advantage of which topology?

- A. Star B. Tree C. Bus D. Ring

Section B — Fill in the Blanks

21. SMTP stands for_____.

22. _____ is preferably used in the Internet Backbone of Country.

23. _____ communication technology delivers both voice and multimedia communication over the internet protocol.

24. _____ communication technique establishes a dedicated communication path between two devices for the entire duration of a transmission, ensuring a continuous and consistent connection?

25. _____ is a set of rules that needs to be followed by the communicating parties in order to have successful and reliable data connection over the network.

Section C — True/False

26. A set of rules and regulations that governs the Internet is called Protocol.

27. A repeater handles different protocols.

28. A switch is a device used to segment networks into subnets.

29. A location on a net server is called a website.

30. A document that uses HTTP is called a webpage.

Section D - Assertion and Reason Based Questions

31. Assertion (A): Star topology is more reliable than bus topology.

Reason (R): In star topology, each device is connected to a central hub, reducing dependence on a single cable.

32. Assertion (A): In ring topology, data moves in multiple directions.

Reason (R): Ring topology has multiple paths between nodes.

33. Assertion (A): SMTP is used to receive emails.

Reason (R): SMTP is a sending protocol, not a receiving protocol.

34. Assertion (A): Switch operates at the Data Link Layer of the OSI model.

Reason (R): Switch uses MAC addresses to forward data.

35. Assertion (A): In LAN, devices are spread over a large geographical area.

Reason (R): LAN is designed for short distances like within a building.

36. Assertion (A): A protocol is the set of rules that are applicable for data sharing and communication over a network.

Reason (R): LAN is an example of Protocol.

37. Assertion (A): Repeater is a device that amplifies the network over geographical distance.

Reason (R): Hub is a device which is used to connect more than one device in the network.

38. Assertion (A): VoIP stands for Voice Over Internet protocol

Reason (R): VoIP is a technology that allows us to make voice calls using a broadband connection.

39. Assertion (A): A router is more reliable and intelligent device than a hub or switch.

Reason (R): Router has advanced capabilities as it can analyze the data being sent over a network and decide how it is packaged before sending to another network.

40. Assertion (A): Bus topology is based on a central node which acts as a hub.

Reason (R): Bus topology installation involves low cost.

2 Marks Questions

1. A. What is the difference between World Wide Web & Internet?

OR

B. What is the difference between domain name and IP address?

2. A. Expand the following: I. POP II. HTTPS

B. Write one advantage and one disadvantage of star topology.

3. A. Define the following terms: I. Modem II. Gateway

OR

B. I. Expand the following terms: HTTP and FTP

II. Differentiate between web server and web browser

4. A. Expand the following:

I. SMTP II. VoIP

B. Give one disadvantage of Star topology

OR

A. What is a web browser?

B. Define the term Telnet

5. A. Define the following terms: I. Router II. Bridge

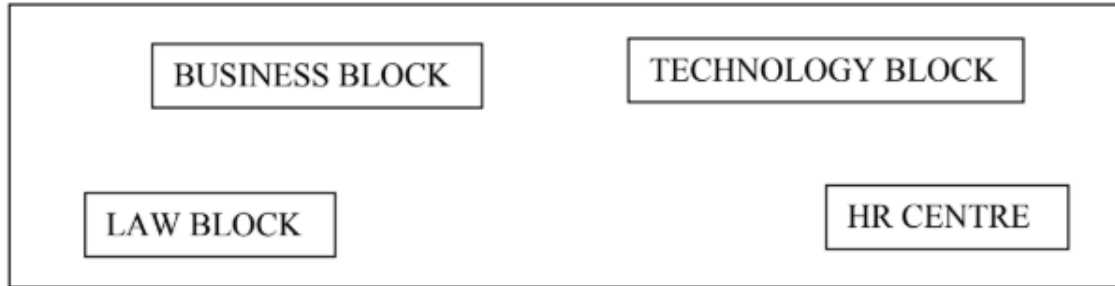
OR

B. I. Expand the following terms: POP3 and ARPANET

II. Differentiate between HTML and XML.

5 marks Questions

1. MyPace University is setting up its academic blocks at Naya Raipur and is planning to set up a network. The University has 3 academic blocks and one Human Resource Center as shown in the diagram below:



Center to Center distances between various blocks/center is as follows :

Law Block to business Block	40m
Law block to Technology Block	80m
Law Block to HR center	105m
Business Block to technology Block	30m
Business Block to HR Center	35m
Technology block to HR center	15m

Number of computers in each of the blocks/Center is as follows :

Law Block	15
Technology Block	40
HR center	115
Business Block	25

Write the Answers:

- I. Suggest the most suitable place (i.e., Block/Center) to install the server of this University with a suitable reason.
- II. Suggest an ideal layout for connecting these blocks/centers for a wired connectivity.
- III. Which device will you suggest to be placed/installed in each of these blocks/centers to efficiently connect all the computers within these blocks/centers
- IV. Suggest the placement of a Repeater in the network with justification.
- V. The university is planning to connect its admission office in Delhi, which is more than 1250km from university. Which type of network out of LAN, MAN, or WAN will be formed? Justify your answer.

2. XYZNova Inc. is planning a new campus in Hyderabad while maintaining its headquarters in Bengaluru. The campus will have four buildings: HR, Finance, IT, and Logistics. As a network expert, you are tasked with proposing the best network solutions for their needs based on the following:

From	To	Distance (in meters)
HR	Finance	50
HR	IT	175
HR	Logistics	90
Finance	IT	60
Finance	Logistics	70
IT	Logistics	60

Number of Computers in Each Block:

Block	Number of Computers
HR	60
Finance	40
IT	90
Logistics	35

- I. Suggest the best location for the server in the Hyderabad campus and explain your reasoning.
- II. Suggest the placement of the following devices: a) Repeater b) Switch III. Suggest and draw a cable layout of connections between the buildings inside the campus.
- IV. The organisation plans to provide a high-speed link with its head office using a wired connection. Which of the cables will be most suitable for this job?
- V. A. What is the use of VoIP?

OR

- B. Which type of network (PAN, LAN, MAN, or WAN) will be formed while connecting the Hyderabad campus to Bengaluru Headquarters?

3. Oxford college, in Delhi is starting up the network between its different wings. There are four Buildings named as SENIOR, JUNIOR, ADMIN and HOSTEL as shown below:

The distance between various building is as follows:

ADMIN TO SENIOR	200 m
ADMIN TO JUNIOR	150 m
ADMIN TO HOSTEL	50 m
SENIOR TO JUNIOR	250 m
SENIOR TO HOSTEL	350 m
JUNIOR TO HOSTEL	350 m

Number of computer in each building is :

SENIOR	130
JUNIOR	80
ADMIN	160
HOSTEL	50

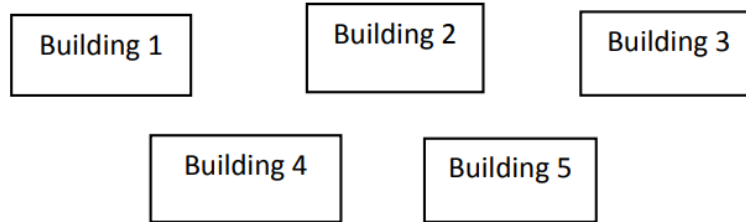
- I. Suggest the cable layout of connections between the buildings.
- II. Suggest the most suitable place (i.e., building) to house the server of this college, provide a suitable reason.
- III. Is there a requirement of a repeater in the given cable layout? Why/ Why not?
- IV. Suggest the placement of hub/switch with justification.
- V. The organisation also has inquiry office in another city about 50-60 km away in hilly region. Suggest the suitable transmission media to interconnect to college and inquiry office out of the following:
 - a. Fibre optic cable b. Microwave c. Radio wave

or

What would be your recommendation for enabling live visual communication between the Admin Office at the Delhi campus and the Mumbai Branch Office from the following options:

- a. Video Conferencing b. Email c. Telephony d. Instant Messaging

4. NeuTech Computers decided to open a new office at Bangalore, the office consist of Five Buildings and each contains number of computers. The details are shown below.



Distance between the buildings –

Building 1 and 2	20 Meters
Building 2 and 3	50 Meters
Building 3 and 4	120 Meters
Building 3 and 5	70 Meters
Building 1 and 5	65 Meters
Building 2 and 5	50 Meters

Building	No. of Computers
1	40
2	45
3	110
4	70
5	60

Computers in each building are networked but buildings are not networked so far. The Company has now decided to connect building also.

I. Suggest a cable layout for connecting the buildings.

II. Do you think anywhere Repeaters required in the campus? Why?

III. The company wants to link this office to their head office at Delhi

a. Which type of transmission medium is appropriate for such a link?

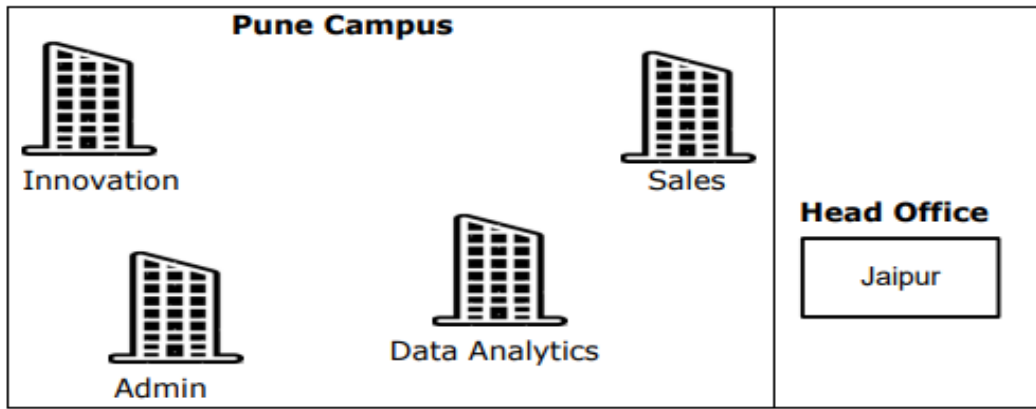
OR

b. What type of network would this connection result into?

IV. Where server is to be installed? Why?

V. Suggest the wired Transmission Media used to connect all buildings efficiently

5. "Tech Innovators" is planning to set up its India campus at Pune with its Head Office at Jaipur. The Pune campus has 4-main blocks-Innovation, Sales, Data Analytics and Admin. You as a network expert have to suggest the best network related solutions for their problems raised in I TO V.



Distance between various building blocks:

Innovation block to Sales block	130 m
Innovation block to Data Analytics block	95 m
Innovation block to Admin block	60 m
Admin block to Data Analytics block	50 m
Admin block to Sales block	125 m
Pune campus to Head office	1202 KM
Data Analytics to sales block	55 m

Number of computers in each block:

Innovation block	20
Sales block	10
Data Analytics	60
Admin	25

- I. Suggest the most appropriate location of the server inside the Pune campus (out of the 4 blocks), to get the best connectivity for maximum number of computers. Justify your answer.
- II. Suggest network type (out of LAN, MAN, WAN) for connecting each of the following set of their offices: a. Pune Campus and Head Office b. Admin block and sales block
- III.. Suggest the placement of the following devices at Pune Campus:
 - a. Switch
 - b. Repeater
- IV. Company wants to connect its local offices situated in Pune. Company can compromise with speed but cost of connectivity should be economic. Which of the following communication medium, you will suggest to be procured by the company:
 - Telephone cable
 - Twisted Pair Cable
 - Optical Fiber Cable

V. A. Which of the following will you suggest to establish the online face to-face communication between the people in the Admin office of Pune Campus and Jaipur Head office?

- a. Cable TV
- b. E-mail
- c. Video Conferencing
- d. Text Chat

OR

B. Suggest and draw the cable layout to efficiently connect various blocks within the Pune campus for connecting the computers

Mr. Chirag Vankar, (PGT CS)
PMSHRI KV Junagadh

DATABASE MANAGEMENT SYSTEM AND CONNECTIVITY

Section A — Multiple Choice Questions

- 1 A table has two candidate keys, one of which is chosen as the primary key. How many alternate keys does this table have?
A. 0 B. 1 C. 2 D. 3
- 2 Which of the following SQL command can change the degree of the existing relation?
A. DROP TABLE B. ALTER TABLE
C. UPDATE...SET D. DELETE
- 3 What will be the output of the query?
SELECT MACHINE_ID, MACHINE_NAME FROM INVENTORY WHERE QUANTITY <= 100;
A. All columns of INVENTORY table with quantity greater than 100
B. ID and name of machines with quantity less than 100 from INVENTORY table
C. All columns of INVENTORY table with quantity greater than or equal to 100
D. ID and name of machines with quantity less than or equal to 100 from INVENTORY table.
- 4 A relation in MySQL database consists of 2 tuples and 3 attributes. If 2 attributes are deleted and 4 tuples are added, what will be the cardinality of the relation?
A. 4 B. 5 C. 6 D. 7
- 5 Which aggregate function in SQL returns the smallest value from a column in a table?
A. MIN() B. MAX() C. SMALL() D. LOWER()
- 6 A relational database consists of a collection of
A. Tuple B. Attribute C. Relation D. Keys

- 7 DML stands for:
- A. Data Manageable Language B. Domain Manipulation Language
C. Data Manipulation Language D. None of the above
- 8 Which of the following is not a DDL command?
- A. CREATE B. UPDATE C. ALTER D. TRUNCATE
- 9 Which of the following is not a DML command?
- A. SELECT B. UPDATE C. DELETE D. CREATE
- 10 A row is also called _____
- A. Field B. Tuple C. Both A and B D. None of the above
- 11 A table can have _____
- A. Many primary keys and many unique keys.
B. One primary key and one unique key.
C. One primary key and many unique keys.
D. Many primary keys and one unique key.
- 12 Which of the following statements will delete all rows in a table namely mytable without deleting the table's structure?
- A. DELETE FROM mytable; B. DELETE TABLE mytable;
C. DROP TABLE mytable; D. None of these.
- 13 Key to represent relationship between tables is called
- A. Primary Key B. Secondary Key
C. Foreign Key D. None of these
- 14 Which clause is used to specify conditions on groups created using aggregate functions?
- A. WHERE B. GROUP BY C. HAVING D. ORDER BY
- 15 Aggregate functions can be used in the SELECT list or the ____ clause of a SELECT statement. They cannot be used in a ____ clause?
- A. WHERE, HAVING B. HAVING, WHERE
C. GROUP BY, HAVING D. GROUP BY, WHERE

Section C - TRUE / FALSE

- 26 The duplication of data is known as data redundancy.
- 27 A primary key can store empty values in it.
- 28 In MySQL, you can change the column name using UPDATE command.
- 29 The SQL keyword GROUP BY instructs the DBMS to group together those rows that have the same value in a column.
- 30 A DELETE, UPDATE, or INSERT query requires commit() to reflect the changes in the database.

Section D - ASSERTIONS AND REASONS

Mark the correct choice as:

- A. Both A and R are true and R is the correct explanation for A
 - B. Both A and R are true and R is not the correct explanation for A
 - C. A is True but R is False
 - D. A is False but R is True
- 31 **Assertion (A):** The PRIMARY KEY constraint in SQL ensures that each value in the column(s) is unique and cannot be NULL.
Reason (R): Candidate keys are not eligible to become a primary key.
- 32 **Assertion (A):** Data redundancy may lead to data inconsistency.
Reason (R): When redundant data or multiple copies of data mismatch, it makes the data inconsistent.
- 33 **Assertion (A):** A primary key is used to uniquely identify the rows in a data table.
Reason (R): A primary key is a field or attribute which has a unique value for each row or tuple.
- 34 **Assertion (A):** A data table can have only one primary key.
Reason (R): In a data table, there can be only one attribute/field containing unique values for each row.

35 **Assertion (A):** Both WHERE and HAVING clauses are used to specify conditions.

Reason (R): Both WHERE and HAVING are interchangeable.

36 **Assertion (A):** The GROUP BY clause is used to group rows having the same values in one or more columns.

Reason (R): The GROUP BY clause is commonly used with aggregate functions such as SUM(), AVG(), and COUNT().

37 **Assertion (A):** The HAVING clause is used to specify conditions on groups created using the GROUP BY clause.

Reason (R): The WHERE clause is used to filter individual rows before grouping takes place.

38 **Assertion (A):** The GROUP BY clause can be used without an aggregate function.

Reason (R): GROUP BY groups rows having the same values in the specified column(s), even if no aggregate function is used.

39 **Assertion (A):** A cursor is required to execute SQL queries from a Python program connected to MySQL.

Reason (R): The cursor provides methods such as execute() for executing SQL statements.

40 **Assertion (A):** The is_connected() method is used to execute an SQL query in MySQL.

Reason (R): The execute() method of the cursor is used to execute an SQL statement.

ANSWERS:

Question No	Answer	Question No	Answer	Question No	Answer	Question No	Answer
1	B	11	C	21	Domain	31	C
2	B	12	A	22	Foreign key	32	A
3	D	13	C	23	Having	33	A
4	C	14	C	24	Group By	34	C
5	A	15	B	25	=	35	C
6	C	16	B	26	True	36	B
7	C	17	C	27	False	37	B
8	B	18	C	28	False	38	A
9	D	19	D	29	True	39	A
10	B	20	B	30	True	40	D

SHORT ANSWER QUESTIONS

1. Meenu has been entrusted with the management of NSE Database. She needs to access some information from STOCK and TRADERS tables for a survey analysis.

Help him extract the following information by writing the desired SQL queries as mentioned below.

Table Name: TRADERS					
TCODE		TNAME		CITY	
T01		RELIANCE DIGITAL		MUMBAI	
T02		TATA DIGITAL		BHUBANESWAR	
T03		BIRLA DIGITAL		NEW DELHI	
Table name: STOCK					
SCODE	SNAME	QTY	PRICE	BRAND	TCODE
1001	COMPUTER	90	45000	DELL	T01
1006	LCD PROJECTOR	40	42000	NEC	T02
1004	IPAD	100	55000	APPLE	T01
1003	DIGITAL CAMERA	160	15000	SAMSUNG	T02
1005	LAPTOP	600	35000	HP	T03

Write SQL queries for the following:

- (i) Display the SNAME, QTY, PRICE, TCODE, and TNAME of all the stocks in the STOCK and TRADERS tables.
- (ii) Display the details of all the stocks with a price ≥ 35000 and ≤ 50000 (inclusive).
- (iii) Display the SCODE, SNAME, QTY*PRICE as the "TOTAL PRICE" of BRAND "NEC" or "HP" in ascending order of QTY*PRICE.
- iv) Display the number of stock items in each TCODE.

OR

To display the Cartesian Product of these two tables

Answers :

- i) select sname, qty, price , stock.tcode, tname from stock, traders where stock.tcode = traders.tcode;
- ii) select * from stock where price between 35000 and 50000;
- iii) select scode, sname, qty * price as "total price" from stock where brand in ('nec', 'hp') order by qty * price asc;
- iv) select tcode, count(*) as stock_count from stock group by tcode;

2. Write SQL Commands for (a) to (e) and write the outputs for (f) on the basis of table:

FURNITURE

NO	ITEM NAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
1	White Lotus	Double Bed	2002-02-23	3000	25
2	Pink feathers	Baby Cot	2002-01-29	7000	20
3	Dolphin	Baby Cot	2002-02-19	9500	20
4	Decent	Office Table	2002-02-01	25000	30
5	Comfort zone	Double Bed	2002-02-12	25000	30
6	Donald	Baby cot	2002-02-24	6500	15
7	Royal Finish	Office Table	2002-02-20	18000	30
8	Royal tiger	Sofa	2002-02-22	31000	30
9	Econo sitting	Sofa	2001-12-13	9500	25
10	Eating Paradise	Dinning Table	2002-12-19	11500	25

- To show all the information about the Baby cots from the furniture table.
- To list the itemname which are priced at more than 15000 from the furniture table.
- To list itemname and type of those items, in which dateofstock is after 2002-02-01 from the furniture table in descending order of itemname
- To display itemname and dateofstock of those items, in which the discount percentage is more than 25 from the furniture table.

Answers :-

- select * from client where city = 'delhi';
- update product set price = price + 10 where pr_name = 'bath soap';
- select * from product where price = (select max(price) from product);
- select p.pr_name, p.price, c.c_name, c.city from product p join client c on p.pr_id = c.pr_id;

DATABASE CONNECTIVITY

1. TO EXECUTE SELECT QUERY WITH WHERE CLAUSE USING A PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system",
database="student")
cur=con.cursor( )
cur.execute("select * from student where marks>90")
r=cur.fetchall( )
count=cur.rowcount
print("total no of rows:",count)
for row in r:
    print(row)
```

2. TO UPDATE A DATA IN A TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system",
database="student")
cur=con.cursor( )
cur.execute("UPDATE STUDENT SET MARKS=100 WHERE MARKS=40")
con.commit( )
print(cur.rowcount,"RECORD UPDATED")
```

3. TO DELETE A RECORD FROM THE TABLE USING PYTHON INTERFACE

```
import mysql.connector as m
con =m.connect(host="localhost",user="root", passwd="system",
database="student")
cur=con.cursor( )
cur.execute("DELETE FROM STUDENT WHERE MARKS<50")
con.commit( )
print(cur.rowcount,"RECORD DELETED")
```

4. Rehaan wants to write a program in Python to insert the following record in the table named EMP in MYSQL database COMPANY:

- a. eno(Empno)- integer
- b. ename(Name) - string
- c. DOB (Date of birth) – Date
- d. Salary – float

Note the following to establish connectivity between Python and MySQL:

Username – root, Password – password, Host - localhost

The values of fields eno, name, DOB and salary has to be accepted from the user.
Help Rehaan to write the program in Python.

Answer :-

```
def Add(mycursor):
```

```
    mycursor.execute('use COMPANY')
```

```
    n = int(input('Enter no. of Records to be entered: '))
```

```
    for i in range(n):
```

```
        en = int(input('Enter Student Roll No: '))
```

```
        enm = input('Enter Student name: ')
```

```
        dob = input('Enter Student DOB as 'YYYY-MM-DD' : '))
```

```
        sal = float(input('Enter Student Fee: '))
```

```
        query = f "insert into EMP values({en},'{enm}','{dob}',{sal})"
```

```
        mycursor.execute(query)
```

```
        mycon.commit( )
```

```
import mysql.connector as m
```

```
mycon=m.connect(host="localhost",user="root",password="password")
```

```
mycursor=mycon.cursor( )
```

```
Add(mycursor)
```

```
mycon.close( )
```

Mr. Mayank Gour, (PGT-CS)

KV VV Nagar



केन्द्रीय विद्यालय संगठन (अहमदाबाद संभाग) KENDRIYA VIDYALAYA SANGATHAN (AHMEDABAD REGION)



Artificial Intelligence
Shaping the Future



pandas
Data Analysis
made easy



Two days
CONTENT ENRICHMENT TRAINING
PROGRAMME - 2026
for PGT - Comp. Sci.
(Ahmedabad Region)

Dates: 21st & 22nd August 2026



DATES
21st & 22nd
August 2026



TIME
09:00 AM to
05:30 PM



MODE
Face-to-Face



FOR
PGT (Computer Science
& Informatics Practices)
of Ahmedabad Region



AI Tools for
Smart Teaching



Python
Programming
Essentials



Data Handling
with Pandas



Hands-on
Practice &
Projects



Enhance Skills &
Classroom Impact



Venue
पी एम श्री केन्द्रीय विद्यालय क्र. 3
PM SHRI KENDRIYA VIDYALAYA NO.3
वायुसेना स्टेशन - 2, जामनगर - 361004
AIR FORCE STATION - II, JAMNAGAR - 361004 (GUJARAT)



LEARN
Keep Learning



INSPIRE
Ignite Curiosity



INNOVATE
Think Beyond



EMPOWER
Create Impact

Empowering Educators,
Building a Brighter Tomorrow!