12th Public Exam Answer Key 2019 – 2020

Computer Science

PART – A

1. a) Pure functions

2. d) Tuple

3. b) Private members

4. d) Time and space

5. d) Integrated Development learning Environment

6. c) 2 4 6 8

7. b) 14

8. e) Type Error

9. c) :

10. a) .

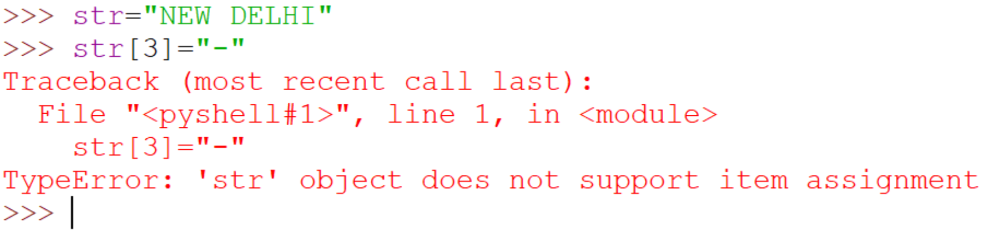
11. b) σ

12. a) DROP

13. d) Flat File

14. b) Boost

15. d) Database



```
>>> str="NEW DELHI"
>>> str[3]="-"
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    str[3]="-"
TypeError: 'str' object does not support item assignment
>>> |
```

"Strings are immutable in 'Python',
which means you can't make any changes once you declared"

Ref. Pg.no : 116 and 117 (First Example)

PART – B

16. Pair:

- Any way of bundling **two** values **together** into **one** can be considered as a **pair**.
- Pair is a **compound structure** which is made up of **list or Tuple**

Example:

lst[(0, 10), (1, 20)] here ,**(1, 20) (0, 10)** - are **pairs**

17. Namespaces:

- Namespaces are **containers** for mapping names of **variables** to **objects**.
- Programming languages keeps track of all mappings with namespaces.

Example: (name : = object). **a:=5**

18. Algorithm:

- An algorithm is a finite set of instructions to accomplish a particular task.
- It is a step-by-step procedure for solving a given problem.

19. range():

- range() generates a list of values starting from start till stop-1 in for loop.
- The syntax of range() is as follows:

range (start, stop, [step])

Where, **start** – refers to the initial value
stop – refers to the final value
step – refers to increment value, this is optional part.

20. Categories of SQL commands:

- DML - Data Manipulation Language
- DDL - Data Definition Language
- DCL - Data Control Language
- TCL - Transaction Control Language
- DQL - Data Query Language

21. Expansion:

SWIG	→ Simplified Wrapper Interface Generator
MinGW	→ Minimalist GNU for Windows

22. Advantage of "INTEGER PRIMARY KEY":

- If a column of a table is declared to be an **INTEGER PRIMARY KEY**, then whenever a NULL will be used as an input for this column, the **NULL will be automatically converted into an integer** which will one larger than the highest value so far used in that column.
- If the table is empty, the value 1 will be used.

23. Types of Data Visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

24. Output:

a) COMPUTER SCIENCE COMPUTER SCIENCE b) COMPUTE

PART – C

25. Difference Between Pure and Impure function:

PURE FUNCTION	IMPURE FUNCTION
➢ Pure functions will give exact result when the same arguments are passed.	➢ Impure functions never assure you that the function will behave the same every time it‟s called.
➢ Pure function does not cause any side effects to its output.	➢ Impure function causes side effects to its output.
➢ The return value of the pure functions solely depends on its arguments passed.	➢ The return value of the impure functions does not solely depend on its arguments passed.
➢ They do not modify the arguments which are passed to them.	➢ They may modify the arguments which are passed.
➢ **Example**: strlen(), sqrt()	➢ **Example**: random(), Date()

26. Asymptotic Notation:

- **Asymptotic Notations** are languages that use meaningful statements about time and space complexity.
- The following three asymptotic notations are mostly used to represent time complexity of algorithms:

(i) Big O

- Big O is often used to describe the worst-case of an algorithm.

(ii) Big Ω

- Big Omega is the reverse Big O

Example:

If **Big O** is used to describe the upper bound (worst - case) then, **Big Ω** is used to describe the lower bound (best-case).

(iii) Big Θ

- When an algorithm has a complexity with **lower bound = upper bound**, that algorithm has a complexity O (n log n) and Ω (n log n), it's actually has the complexity Θ (n log n).
- Time complexity is **n log n** in both best-case and worst-case.

27. Ternary Operator:

- Ternary operator is also known as **conditional operator** that evaluates something based on a condition being true or false.
- It simply allows testing a condition in a single line replacing the multiline if-else making the code compact.

SYNTAX*:* ***Variable Name = [on_true] if [Test expression] else [on_false]***

Example: min = 20 if 49<50 else 60 # Output**: min = 20**

28. Recursive Function:

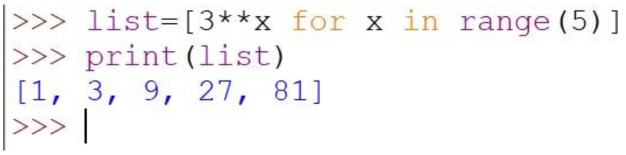
- Functions that calls itself is known as recursive.

Works:

- Recursive function is called by some external code.
- If the base condition is met then the program gives meaningful output and exits.
- Otherwise, function does some required processing and then calls itself to continue recursion.

29. Output:

[1, 3, 9, 27, 81]



```
>>> list=[3**x for x in range(5)]
>>> print(list)
[1, 3, 9, 27, 81]
>>> |
```

30. TCL Commands:

- Transactional control language (TCL)is used to manage transactions in the database.
- It is to manage the changes made to the data in a table.

TCL command:

Commit: **Saves** any transaction into the database **permanently**

Rollback: **Restores** the database to last **commit** state

Savepoint: **Temporarily** save a transaction so that you can rollback

31. Difference between reader and DictReader:

Reader()	DictReader()
➢ **csv.reader()** - to read CSV file in to **list/tuple**..	➢ **DictReader()** - To read a CSV file into a dictionary
➢ It will take each line of the file and make a list of all columns.	➢ It works similar to the reader() class
➢ By this method one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (,).	➢ It creates an object which maps data to a dictionary.
➢ syntax: **csv.reader(fileobject,delimiter,fmtparams)**	➢ The keys are given by the fieldnames as parameter.

32. **Difference between fetchone() and fetchmany():**

fetchone()	fetchmany()
➢ The **fetchone()** method returns the next row of a query result set or None in case there is no row left	➢ The **fetchmany()** method returns the next number of rows (n) of the result set.
➢ Using while loop and fetchone() method we can display all the records from a table.	➢ Displaying specified number of records is done by using **fetchmany().**

33. **Output:**

Welcome to Python Programming

PART – D

34. a) **Linear Search:**

Linear search also called sequential search is a sequential method for finding a particular value in a list. This method checks the search element with each element in sequence until the desired element is found or the list is exhausted. In this searching algorithm, list need not be ordered.

Pseudo code

```
1. Traverse the array using for loop
2. In every iteration, compare the target search key value with the current value of the list.
     ✓ If the values match, display the current index and value of the array
     ✓ If the values do not match, move on to the next array element.
3. If no match is found, display the search element not found.
```

Example:

Input: values[] = {5, 34, 65, 12, 77, 35}
target = 77
Output: 4

34. b) **Input and Output function:**

- The **input()** function helps to **enter data** at **run time** by the user
- In Python, the **print()** function is used to display result on the screen.

input() functions with examples

- In Python, **input( )** function is used to accept data as input at run time.
- Syntax is **Variable =input ("prompt string")**
  - **prompt string** is used, to **display statement** or **message** on the monitor.
- input( ) with prompt string
  - **>>>city=input ("Enter Your City: ")**
    - Enter Your City: chennai

- input( ) without prompt string
  - **>>>city=input()**
- The **input ( )** accepts all data as **string or characters** but **not as** numbers.
- The **int( )** function is used to convert string data as integer data explicitly.

print() functions with examples

- In Python, the **print()** function is used to display result on the screen.
- The syntax for **print()** is as follows:
  - **print (“string to be displayed as output ”)**
  - **print (variable)**
  - **print (“String to be displayed as output ”, variable)**
  - **print (“Str1 ”, var1, “Str 2”, var2,“Str 3” ……)**

```
>>>print (“Welcome to Python”)
        Welcome to Python
>>> x = 5
>>>y = 10
>>>print (x)
        5
>>> print (“The No is = ”, x)
        The No is =5
>>> print (“The sum of ”, x, “ and ”, y, “ is ”, x+y)
        The sum of 5 and10 is 15
```

35. a) i) **Display all 3 digit even Number:**

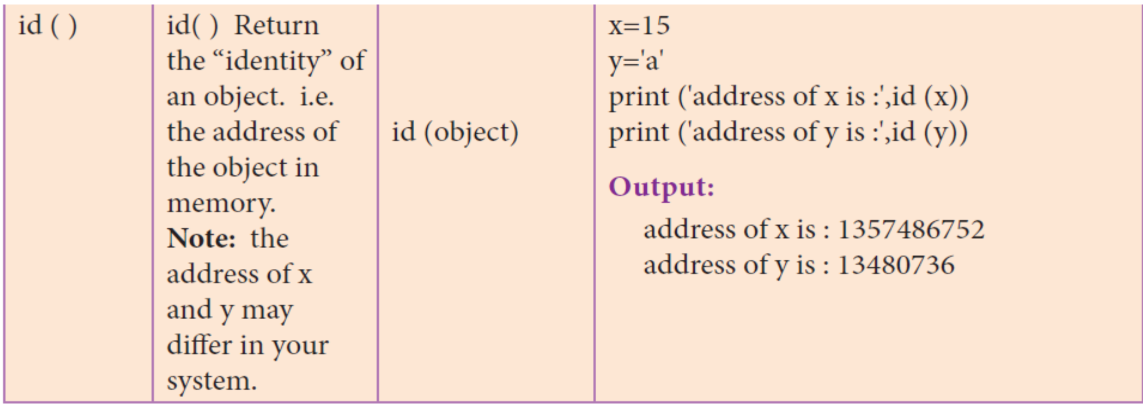
```
for I in range(100,999,2):
        print(i”,”)
```

O/P: 100,102,104,…….998

ii) **Output:**

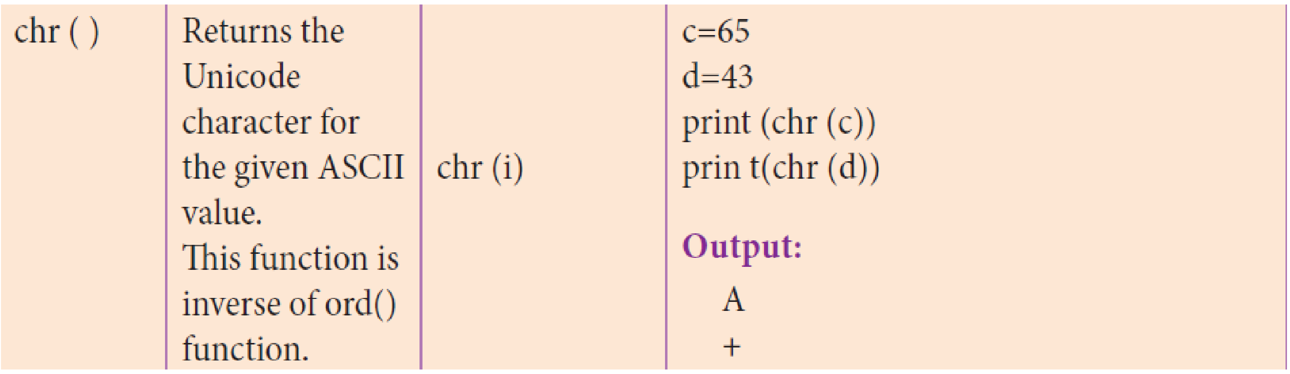
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

35. b) i)



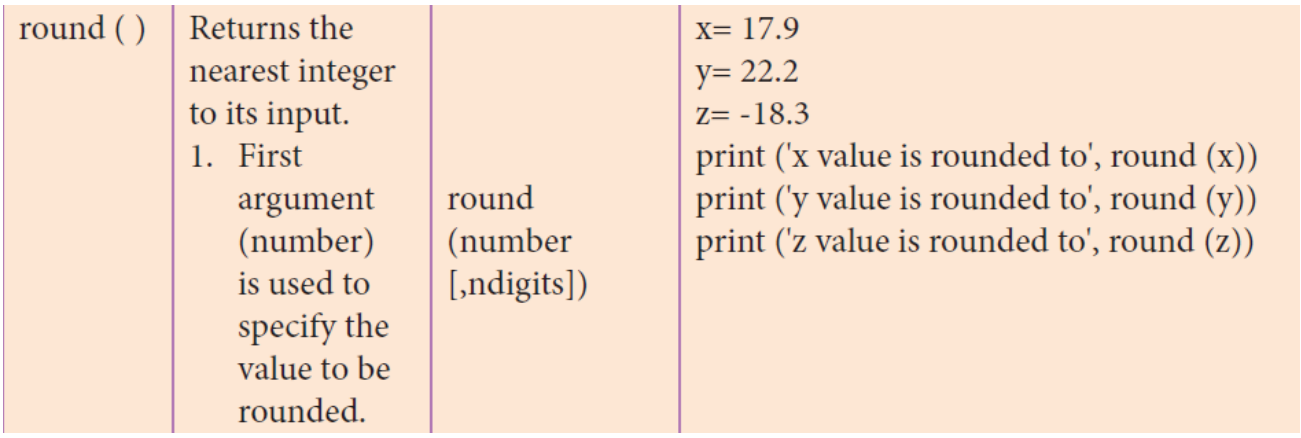
id ( )	id( ) Return the “identity” of an object. i.e. the address of the object in memory. **Note:** the address of x and y may differ in your system.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **Output:** address of x is : 1357486752 address of y is : 13480736

ii)



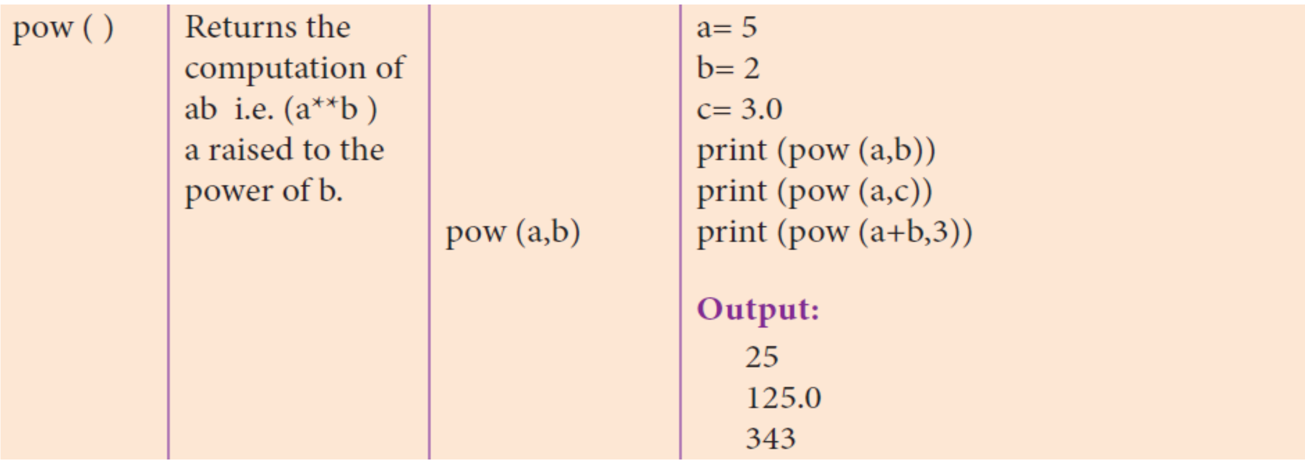
chr ( )	Returns the Unicode character for the given ASCII value. This function is inverse of ord() function.	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) **Output:** A +

iii)



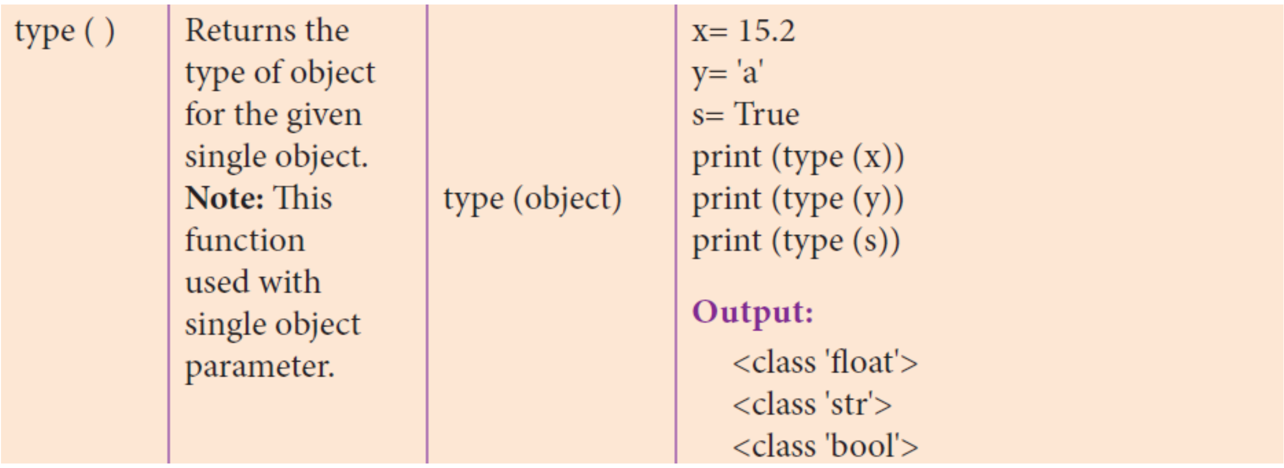
round ( )	Returns the nearest integer to its input.
	1. First argument (number) is used to specify the value to be rounded.
		round (number [,ndigits])	x= 17.9
			y= 22.2
			z= -18.3
			print ('x value is rounded to', round (x))
			print ('y value is rounded to', round (y))
			print ('z value is rounded to', round (z))

iv)



pow ( )	Returns the computation of ab i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) **Output:** 25 125.0 343

v)



type ( )	Returns the type of object for the given single object. **Note:** This function used with single object parameter.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) **Output:** <class 'float'> <class 'str'> <class 'bool'>

36. a) **Output:**

```
i) Welcome to Python

ii) Python

iii) Pto

iv) Wotyn

v) nytoW
```

36. b) **Constructor:**

- **"init"** is a special function begin and end with double underscore in Python act as a Constructor.
- Constructor function will automatically executed when an object of a class is created.

General format:

```
def __init__(self, [args ……..]):
        <statements>
```

Example:

```
class saro:
        def __init__(self):
                a="Hi.. "
        obj=saro()
```

Destructor

- Destructor is also a special method gets executed automatically when an object exit from the scope.
- In Python, __del__( ) method is used as destructor.
- It **removes** the memory of an object

General format:

```
def __del__(self):
        <statements>
```

37. a) **Set:**

- A Set is a mutable and an unordered collection of elements without duplicates.

Set Operations:

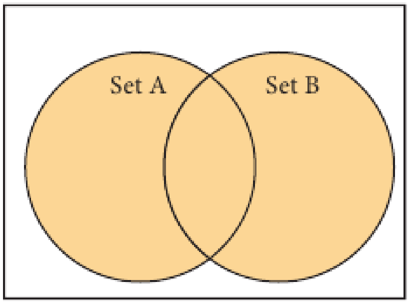
i. Union,
ii. Intersection,
iii. Difference
iv. Symmetric difference

(i) Union:

- ✓ It includes all elements from two or more sets.
- ✓ The **operator |** is used to union of two sets.
- ✓ The function union( ) is also used to join two sets in python.

Example:

```
set_A={2,4,6,8}

set_B={'A', 'B', 'C', 'D'}

U_set=set_A|set_B

print(U_set)
```



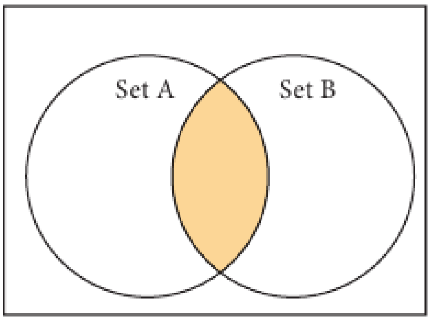
Output: {2, 4, 6, 8, 'A', 'D', 'C', 'B'}

(ii) Intersection:

- ✓ It includes the common elements in two sets.
- ✓ The **operator &** is used to intersect two sets in python.
- ✓ The function **intersection( )** is also used to intersect two sets in python.

Example:

```
set_A={'A', 2, 4, 'D'}

set_B={'A', 'B', 'C', 'D'}

print(set_A & set_B)
```



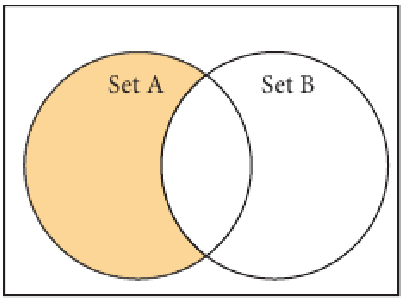
Output: {'A', 'D'}

(iii) Difference:

- ✓ It includes all elements that are in first set (say set A) but not in the second set (say set B).

- ✓ The minus **(-) operator** is used to difference set operation in python.
- ✓ The function **difference( )** is also used to difference operation.

Example:



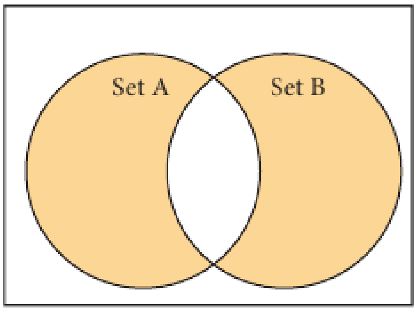
```
set_A={'A', 2, 4, 'D'}

set_B={'A', 'B', 'C', 'D'}

print(set_A - set_B)
```

Output: {2, 4}

(iv) Symmetric difference

- ✓ It includes all the elements that are in two sets (say sets A and B) but not the one that are common to two sets.
- ✓ The caret **(^) operator** is used to symmetric difference set operation in python.
- ✓ The function **symmetric_difference( )** is also used to do the same operation.

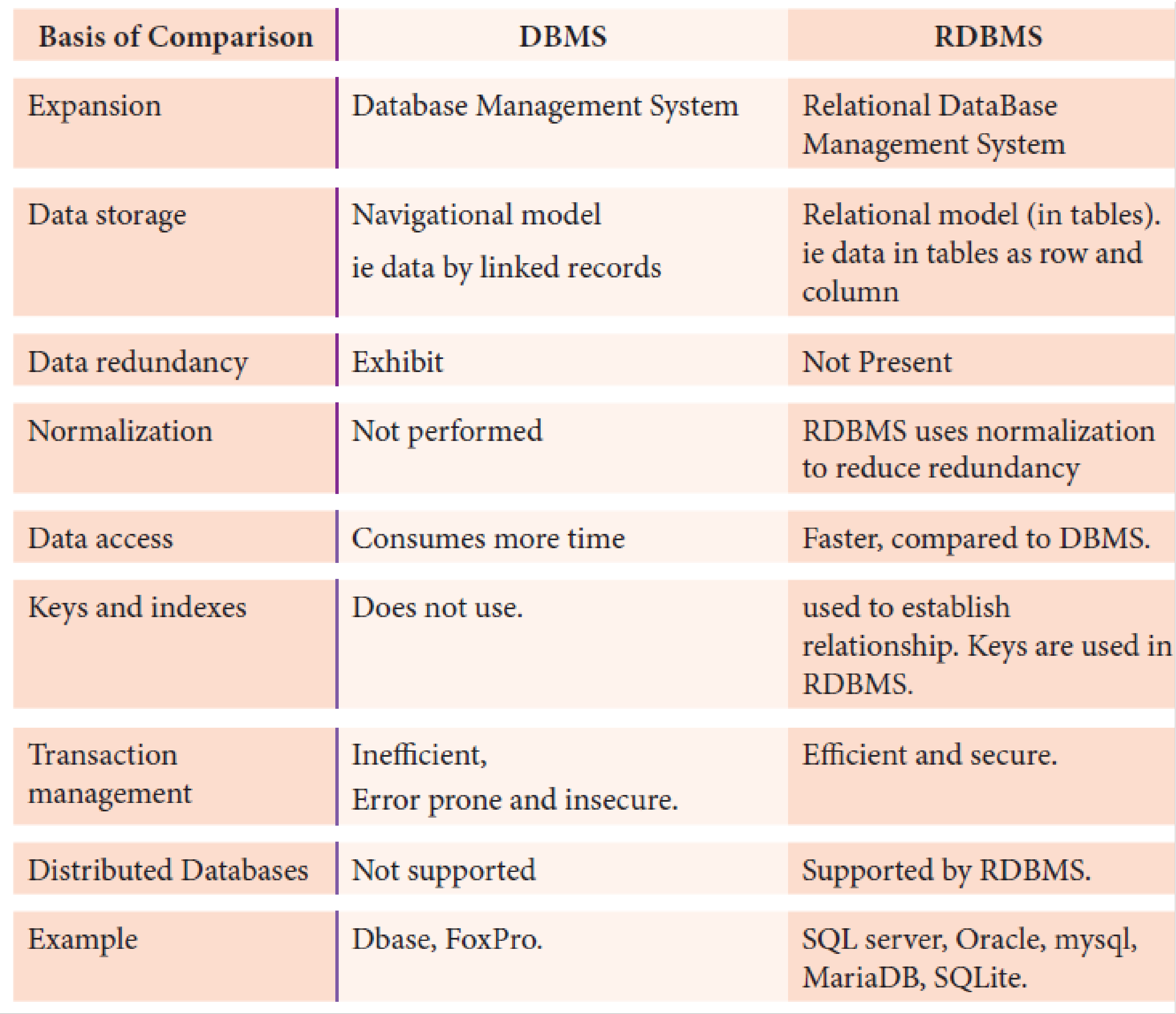
Example:



```
set_A={'A', 2, 4, 'D'}

set_B={'A', 'B', 'C', 'D'}

print(set_A ^ set_B)
```

Output: {2, 4, 'B', 'C'}

37. b) Difference between DBMS and RDBMS:



Basis of Comparison	DBMS	RDBMS
Expansion	Database Management System	Relational DataBase Management System
Data storage	Navigational model ie data by linked records	Relational model (in tables). ie data in tables as row and column
Data redundancy	Exhibit	Not Present
Normalization	Not performed	RDBMS uses normalization to reduce redundancy
Data access	Consumes more time	Faster, compared to DBMS.
Keys and indexes	Does not use.	used to establish relationship. Keys are used in RDBMS.
Transaction management	Inefficient, Error prone and insecure.	Efficient and secure.
Distributed Databases	Not supported	Supported by RDBMS.
Example	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite.

38. a) Program:

```
CREATE TABLE employee (
        empno integer NOT NULL,
        Firstname char(20),
        Lastname char(20),
        Gender char(1),
        Age integer,
        Place char(10),
        PRIMARY KEY (Firstname, Lastname));        → Table constraint
```

38. b) **Features of Python over C++**

- ✓ Python uses Automatic Garbage Collection whereas C++ does not.
- ✓ C++ is a statically typed language, while Python is a dynamically typed language.
- ✓ Python runs through an interpreter, while C++ is pre-compiled.
- ✓ Python code tends to be 5 to 10 times shorter than that written in C++.
- ✓ In Python, there is no need to declare types explicitly where as it should be done in C++
- ✓ In Python, a function may accept an argument of any type, and return multiple values without any kind of declaration beforehand. Whereas in C++ return statement can return only one value.