List

1. Write a Python program to take list int numbers from users and returns a new list containing only the prime numbers from the original list.

2. Implement a function that takes a nested list and returns a flat list containing all the elements. For example, if given **[[1, 2], [3, [4, 5]], 6]**, the output should be **[1, 2, 3, 4, 5, 6]**.

3. Given the names and grades for each student in a class of N students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

   Note: If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

   Example

   records = [["chi", 20.0]], ["beta", 50.0], ["alpha", 50.0]

   The ordered list of scores is [20.0, 50.0], so the second lowest score is 50.0. There are two students with that score: ["beta", "alpha"]. Ordered alphabetically, the names are printed as:

   alpha

   beta

4. Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. You are given n scores. Store them in a list and find the score of the runner-up.
   Explaination: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum score is 5. Hence, we print 5 as the runner-up score.

5. Write a Python program to takes a list of words from user and return the longest word and the length of the longest one.
   Sample Output:
   Longest word: s
   Length of the longest word: 9

6. Write a Python program that takes a list of integers, squares each element, filters out numbers greater than 50, and then returns the sum of the remaining squared values.
   Ex: input_numbers = [3, 6, 8, 10, 12, 7, 5]
   Square_of_num: [9,36,64,100,144,49,25]
   squared_and_filtered : [9, 36, 49, 25]
   Sum of squared values less than or equal to 50: 119

7. Implement a Python function that rotates a given list to the right by a specified number of positions. Additionally, create a program that uses this function to rotate a list and then prints both the original and rotated lists.
   Original List: [1, 2, 3, 4, 5]
   Rotated List (2 positions to the right): [4, 5, 1, 2, 3]

String

1. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog.

   Your task here is to write a function to check a sentence to see if it is a pangram or not.

2. Write a version of a palindrome recognizer that also accepts phrase palindromes such as "Go hanga salami I'm a lasagna hog.", "Was it a rat I saw?", "Step on no pets", "Sit on a potato pan, Otis", "Lisa Bonet ate no basil", "Satan, oscillate my metallic sonatas", "I roamed under it as a tired nude Maori", "Rise to vote sir", or the exclamation "Dammit, I'm mad!". Note that punctuation, capitalization, and spacing are usually ignored.

3. Write a program that takes a string with multiple words as input and capitalizes the first letter of each word. For example, if the input is "hello world," the program should output "Hello World."

4. Create a Python program that calculates and prints the total number of words in a given string. Words are separated by spaces. For example, if the input is "This is a sample sentence," the program should output "The total number of words is: 5."

5. Write a program that takes a string as input and prints a table showing the frequency of each character in the string. Exclude spaces and consider letter casing (e.g., 'A' and 'a' should be counted separately).

6. In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals. ROT-13 ("rotate by 13 places") is a widely used example of a Caesar cipher where the shift is 13. In Python, the key for ROT-13 may be represented by means of the following dictionary:key = {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u', 'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b', 'p':'c', 'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k', 'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S', 'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A', 'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I', 'W':'J', 'X':'K', 'Y':'L', 'Z':'M'} Your task in this is to implement an encoder/decoder of ROT-13. Once you're done, you will be able to read the following secret message:

   Pnrfnepvcure? V zhpucersrePnrfnefnynq! Note that since English has 26 characters, your ROT-13 program will be able to both encode and decode texts written in English.

7. Write a Python program that formats a list of names as a comma-separated string with an "and" before the last name. For example, if the input is ["Alice", "Bob", "Charlie"], the program should output "Alice, Bob, and Charlie."

8. Create a program that takes a sentence as input and reverses the order of words while keeping the words themselves unchanged. For example, if the input is "Hello world," the program should output "world Hello."

9. Write a Python program that takes a phrase as input and generates its acronym. The program should capitalize the first letter of each word in the acronym. For example, if the input is "Artificial Intelligence," the program should output "AI."

10. Write a Python program to get a string made of the first 2 and last 2 characters of a given string. If the string length is less than 2, return the empty string instead.

    Sample String : 'w3resource'
    Expected Result : 'w3ce'
    Sample String : 'w3'
    Expected Result : 'w3w3'
    Sample String : ' w'
    Expected Result : Empty String

11. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.
    Sample String : 'restart'
    Expected Result : 'resta$t'

Loops (While and for)

1. Create a Python program that prints numbers from 1 to 100, but for multiples of 3, print "Fizz," for multiples of 5, print "Buzz," and for multiples of both 3 and 5, print "FizzBuzz."

2. Write a Python program that generates the first **n** numbers in the Fibonacci sequence using a **for** loop. Ensure that the program efficiently computes the sequence.

3. Create a Python program that uses nested loops to print a pyramid pattern with a specified number of rows. For example, if the input is 4, the program should print:

```
             1

           232
```

```
    34543
```

Write a program that takes a number n as input from the user and generates the first n terms of the series formed by squaring the natural numbers.

Sample output

```
Enter a number: 6
The first 6 terms of the series are:
1 4 9 16 25 36
```

8. Write a program that prompts the user to input a number and prints its multiplication table.

9. Write a Python program to print the first 8 terms of an arithmetic progression starting with 3 and having a common difference of 4.
The program should output the following sequence:

```
3 7 11 15 19 23 27 31
```

10. Write a Python program to print the first 6 terms of a geometric sequence starting with 2 and having a common ratio of 3.
The program should output the following sequence:

```
2 6 18 54 162 486
```

11. Write a program that asks the user for a positive integer value. The program should calculate the sum of all the integers from 1 up to the number entered. For example, if the user enters 20, the loop will find the sum of 1, 2, 3, 4, ... 20.

12. write a program that takes a positive integer N as input and calculates the sum of the reciprocals of all numbers from 1 up to N. The program should display the final sum. Output of the program should be like:

```
Enter a positive integer: 5
The sum of reciprocals from 1 to 5 is: 2.28
```

13. Write a program that prompts the user to enter a number and repeats this process 5 times. The program should accumulate the numbers entered and then display the final running total.

Sample Output:

```
Enter a number: 10
Enter a number: 15
Enter a number: 35
Enter a number: 40
Enter a number: 50
The final running total is: 150
```

14. Write a program that prompts the user to enter a positive integer and calculates its factorial. The factorial of a positive integer 'n' is denoted as 'n!'

and is calculated by multiplying all the integers from 1 to 'n' together. For example, the factorial of 5 (denoted as 5!) is calculated as 1 x 2 x 3 x 4 x 5.

The program should display the factorial value if the input is a positive number, or display a message stating that the factorial does not exist for negative numbers. Additionally, for an input of zero, the program should output that the factorial of 0 is 1.

15. Write a Python program that prompts the user to enter a base number and an exponent, and then calculates the power of the base to the exponent. The program should not use the exponentiation operator (**) or the math.pow() function. The program should handle both positive and negative exponents.

1. Write a Python program that prompts the user to enter a positive integer. Your program should display all the factors of the number. Additionally, calculate and display the sum of its factors.

Sample output:

```
Enter a positive integer: 45
Factors: 1 3 5 9 15 45
Sum of factors: 78
```

2. Write a program that uses a loop to repeatedly ask the user to enter positive numbers. The loop will come to an end when a negative number is entered. After collecting all the positive numbers, the program will compute their sum and display the result to the user.

3. Write a program that uses a loop to repeatedly ask the user to enter integers. The loop will come to an end when zero is entered. After collecting all the integers, the program will compute and display the average of all the entered numbers.

4. Write a program to enter the numbers till the user wants and at the end it should display the count of positive, negative and zeros entered.

5. Write a program to enter the numbers till the user wants and at the end the program should display the largest and smallest numbers entered.

6. Write a program that asks the user to input a positive integer. Your program should find and display the sum of digits of number. For example, sum of digits of number 32518 is 3+2+5+1+8 = 19.

7. An Armstrong number of three digits is an integer in which the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3 = 371$.

Write a program to check whether a number is an Armstrong number or not.

8. Write a program that prompts the user to input a number and reverse its digits. For example, the reverse of 12345 is 54321; reverse of 5600 is 65.

9. A palindromic number is a number that remains the same when its digits are reversed. For example, 16461. Write a program that prompts the user to input a number and determine whether the number is palindrome or not.

10. Write a program that prompts the user to input a decimal integer and display its binary equivalent.

11. Write a program that prompts the user to input a binary number and display its decimal equivalent.

12. Write a program that prompts the user to input a positive integer. It should then output a message indicating whether the number is a prime number. A prime number is a number that is evenly divisible only by itself and 1. For example, the number 5 is prime because it can be evenly divided only by 1 and 5. The number 6, however, is not prime because it can be divided evenly by I, 2, 3, and 6.

13. Write a program to obtain the first 25 numbers of a Fibonacci sequence. In a Fibonacci sequence the sum of two successive terms gives the third term. Following are the first few terms of the Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34 55 89...

14. Write a program that prompts the user to input two numbers and display its HCF. The Highest Common Factor (HCF) also called the Greatest Common Divisor (GCD) of two whole numbers, is the largest whole number that's a factor of both of them.

15. Write a program to add first seven terms of the following series using a for loop:

$$\frac{1}{1!}+\frac{2}{2!}+\frac{3}{3!}+\cdots$$

16. Compute the sum up to n terms in the series
1 - 1/2 + 1/3 - 1/4 + 1/5 -... 1/n
where n is a positive integer and input by user.

17. Write programs to print following patterns :

1.

****

****

****

****

2.

*

**

***

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

3.

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

4.

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

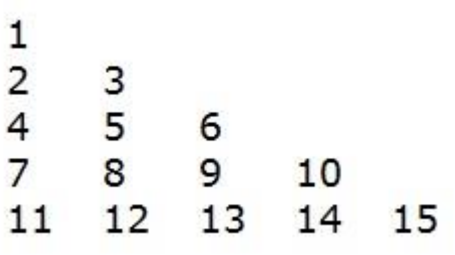
5.

```
    1
   222
  33333
 4444444
555555555
```

6.

```
    1
   212
  32123
 4321234
543212345
```

18. Floyd's triangle is a right-angled triangular array of natural numbers as shown below:



```
1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
```

Write a program to print the Floy'd triangle.

21. Write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." The program should use a loop that repeats until the user correctly guesses the random number. Program should count and display number of tries to win the game.

Control Statements

Write a Python program in which the user enters either 'A', 'B', or 'C'. If 'A' is entered, the program should display the word 'Apple'; if 'B' is entered, it displays 'Banana'; and if 'C' is entered, it displays 'Coconut'.

Write a Python program in which a student enters the number of college credits earned. If the number of credits is greater than 90, 'Senior Status' is displayed; if greater than 60, 'Junior Status' is displayed; if greater than 30, 'Sophomore Status' is displayed; else, 'Freshman Status' is displayed.

Write a program containing a pair of nested while loops that displays the integer values 1–100, ten numbers per row, with the columns aligned as shown below,

```
 1 2 3 4 5 6 7 8 9 10
 11 12 13 14 15 16 17 18 19 20
 21 22 23 24 25 26 27 28 29 30
 .
.
 91 92 93 94 95 96 97 98 99 100
```

File Handling

1) find no of characters, no of words or number of lines in Python text file.

2) Write a program to Count the number of upper-case alphabets present in a text file “PYTHON.TXT”.

Ex: We Learn Python. 3 isupper()

3) A text file “PYTHON.TXT” contains alphanumeric text. Write a program that reads this text file and writes to another file “PYTHON1.TXT” entire file except the numbers or digits in the file.

4) Write a program to count the words “to” and “the” present in a text file “python.txt”.

5) Write a program to display all the lines in a file “python.txt” along with line/record number.

OOPS Example

Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case

1. Create a Class:

a. Define a class called "Student" with attributes such as name, age, and grade.

b. Implement a constructor method that initializes these attributes.

c. Include a method called "display_info()" that prints the details of the student.

2. Instantiate Objects:

a. Create three instances of the Student class, each representing a different student.

b. Assign appropriate values to the attributes of each student object.

c. Call the "display_info()" method for each student object to display their details

a. Create three instances of the Student class, each representing a different student.

b. Assign appropriate values to the attributes of each student object.

c. Call the "display_info()" method for each student object to display their details.

3. Implement Inheritance:

a. Create a subclass called "HighSchoolStudent" that inherits from the Student class.

b. Add an additional attribute to the HighSchoolStudent class called "graduation_year."

c. Implement a constructor method for the HighSchoolStudent class that initializes the attributes of both the Student and HighSchoolStudent classes.

d. Override the "display_info()" method in the HighSchoolStudent class to include the graduation year information.

4. Instantiate HighSchoolStudent Objects:

a. Create two instances of the HighSchoolStudent class, each representing a different high school student.

b. Assign appropriate values to the attributes of each high school student object.

c. Call the "display_info()" method for each high school student object to display their details, including the graduation year.

5. Demonstrate Polymorphism:

a. Create a function called "register_course(student)" that takes a Student object as a parameter and prints a message indicating the student's course registration.

b. Call the "register_course()" function and pass both Student and HighSchoolStudent objects to demonstrate polymorphism.

6. Implement Encapsulation:

a. Add private attributes to the Student class, such as "student_id" and "GPA."

b. Implement appropriate getter and setter methods to access and modify these private attributes.

7. Create Additional Methods:

a. Add a method to the Student class called "study()" that prints a message indicating the student is studying.

b. Add a method to the HighSchoolStudent class called "apply_to_college()" that prints a message indicating the student is applying to college.

8. Write a program that allows the user to interactively create student and high school student objects.

9. Write a program to prompt the user to enter the details of each student, including their name, age, grade, and graduation year (for high school students).

10. Write a program to Create the corresponding objects and display their details using the "display_info()" method.

11. Write a program to implement error handling to ensure that appropriate values are entered

for attributes such as age, grade, and graduation year.

12. Write a program to handle any potential exceptions that may occur during user input.

Function

Practical Question: Implement a Function to Calculate the Total Cost of a Shopping Cart

You are tasked with creating a Python function that calculates the total cost of items in a shopping cart. The function should take a list of items and their prices as input and return the total cost. Additionally, you should consider the application of a discount code if available.

Requirements:

1. Create a function named **calculate_total_cost** that takes two parameters: a list of items and a discount code (optional).
2. The list of items should be a list of dictionaries, where each dictionary contains the item name and its price.
3. The function should calculate the total cost by summing the prices of all items in the cart.
4. If a discount code is provided, apply a percentage discount to the total cost.
5. Return the final total cost after applying the discount.

Example:

```
items = [
    {"item": "Shirt", "price": 25.0},
    {"item": "Jeans", "price": 40.0},
    {"item": "Shoes", "price": 35.0},
]

discount_code = "SAVE20"

total_cost = calculate_total_cost(items, discount_code)
print(f"Total cost after applying the discount: ${total_cost:.2f}")
```

Write the definition of a function zero_ending(scores) to add all those values in the list of scores, which are ending with zero and display the sum.

For example: If the scores contain [200, 456, 300, 100, 234, 678] The sum should be displayed as 600.

6. Write a definition of a method count_now(places) to find and display those place names, in which there are more than 5 characters.

For example :
If the list places contains
["DELHI","LONDON","PARIS","NEW YORK","DUBAI"]
The following should get displayed :
LONDON
NEW YORK

7. Write a method in python to display the elements of list thrice if it is a number and display the element terminated with ‘#’ if it is not a number.

For example, if the content of list is as follows :
ThisList=[‘41’,‘DROND’,‘GIRIRAJ’, ‘13’,‘ZARA’]
The output should be
414141
DROND#
GIRIRAJ#

131313
ZARA#

8. For a given list of values in descending order, write a method in python to search for a value with the help of Binary Search method. The method should return position of the value and should return -1 if the value not present in the list.

9. Write a function half_and_half that takes in a list and change the list such that the elements of the second half are now in the first half.

For example, if the size of list is even and content of list is as follows :
my_liist = [10,20,30,40,50,60]
The output should be
[40,50,60,10,20,30]
if the size of list is odd and content of list is as follows :
my_liist = [10,20,30,40,50,60,70]
The output should be
[50,60,70,40,10,20,30]

10. Write a function that accepts a dictionary as an argument. If the dictionary contains duplicate values, it should return an empty dictionary. Otherwise, it should return a new dictionary where the values become the keys and the keys become the values.

For example, if the dictionary contains the following key-value pairs:
{'a': 10, 'b': 20, 'c': 20}
the function should return an empty dictionary {} because there are duplicate values.

On the other hand, if the dictionary contains the following key-value pairs:
{'a': 10, 'b': 20, 'c': 30}
the function should return a new dictionary {10: 'a', 20: 'b', 30: 'c'} where the values from the origi nal dictionary become the keys, and the keys from the original dictionary become the values.

Dictionary

1. Write a program that reads a string from keyboard and prints the unique words. Your program should convert input string to lower case.

2. Write a program to print all elements in a list those have only single occurrence.
Example: if contents of list is [7, 5, 5, 1, 6, 7, 8, 7, 6].
Your output should be:
1 8

3. Write a program to enter names of employees and their salaries as input and store them in a dictionary.

4. Write a program to read 6 numbers and create a dictionary having keys EVEN and ODD. Dictionary's value should be stored in list. Your dictionary should be like:
{'EVEN':[8,10,64], 'ODD':[1,5,9]}

5. Write a program that reads string from user. Your program should create a dictionary having key as word length and value is count of words of that length. For example, if user enters 'A fat cat is on the mat'.

Word	Word length
A	1
fat	3
cat	3
is	2
on	2
the	3
mat	3

The content of dictionary should be {1:1, 3:4, 2:2}

6. Write a program to input roll numbers and their names of students of your class and store them in the dictionary as the key-value pair. Perform the following operations on the dictionary:
a) Display the Roll numbers and name for all students.
b) Add a new key-value pair in this dictionary and display the modified dictionary
c) Delete a particular student's record from the dictionary
d) Modify the name of an existing students.

Variable, Operator and Expression

1. Write a program that asks the user for his name and then welcomes him. The output should look like this:

```
Enter your name: Saksham
Hello Saksham
```

2. Write a program that prompts the user to enter two integers and display their sum on the screen.

3. Write a program that prompts the user to input a Celsius temperature and outputs the equivalent temperature in Fahrenheit. The formula to convert the temperature is: `F = 9/5 C + 32` where F is the Fahrenheit temperature and C is the Celsius temperature.

4. Write a program which accept principle, rate and time from user and print the simple interest. The formula to calculate simple interest is: `simple interest = principle x rate x time / 100`

5. Write a program that accepts seconds from keyboard as integer. Your program should converts seconds in hours, minutes and seconds. Your output should like this :

```
Enter seconds: 13400
Hours: 3
Minutes: 43
Seconds: 20
```

6. Write a program that prompts the user to enter number in two variables and swap the contents of the variables.

7. Write a program that prompts the user to enter number in two variables and swap the contents of the variables.(Do not declare extra variable.)

8. Write a program that prompts the user to input the radius of a circle and outputs the area and circumference of the circle. The formula is

`Area = pi x radius`^{2}
`Circumference = 2 x pi x radius`

9. Write a program that prompts the user to input the length and the width of a rectangle and outputs the area and perimeter of the rectangle. The formula is

`Area = Length x Width`
`Circumference = 2 x ( Length + Width)`

10. Suppose a, b, and c denote the lengths of the sides of a triangle. Then the area of the triangle can be calculated using the formula:

$$Area = \sqrt{s(s-a)(s-b)(s-c)}$$

where $s = \frac{a+b+c}{2}$

Write a program that asks the user to input the length of sides of the triangle and print the area.

11. Write a program which prompts the user to input principle, rate and time and calculate compound interest. The formula is :

```
CI = P(1+R/100)^T - P
```

SET

- 1: Add a list of elements to a set
- 2: Return a new set of identical items from two sets
- 3: Get Only unique items from two sets
- 4: Update the first set with items that don't exist in the second set
- 5: Remove items from the set at once
- 6: Return a set of elements present in Set A or B, but not both
- 7: Check if two sets have any elements in common. If yes, display the common elements
- 8: Update set1 by adding items from set2, except common items
- 9: Remove items from set1 that are not common to both set1 and set2

Tuple

1: Reverse the tuple

2: Access value 20 from the tuple

3: Create a tuple with single item 50

4: Unpack the tuple into 4 variables

5: Swap two tuples in Python

6: Copy specific elements from one tuple to a new tuple

7: Modify the tuple

8: Sort a tuple of tuples by 2nd item

9: Counts the number of occurrences of item 50 from a tuple

10: Check if all items in the tuple are the same

Conditional Structures

[Set – 1]

1. Write a program that prompts the user to input a number and display if the number is even or odd.

2. Write a Python program that takes an age as input and determines whether a person is eligible to vote. If the age is 18 or above, print "You are eligible to vote." Otherwise, print "You are not eligible to vote yet.".

3. Write a program that prompts the user to input two integers and outputs the largest.

4. Write a program that prompts the user to enter a number and determines whether it is positive, negative, or zero. The program should print "Positive" if the number is greater than 0, "Negative" if the number is less than 0, and "Zero" if the number is 0.

5. Write a program that prompts the user to enter their age and prints the corresponding age group. The program should use the following age groups:

0-12:	Child
13-19:	Teenager
20-59:	Adult
60 and above:	Senior Citizen

6. Write a program that prompts the user to input a number from 1 to 7. The program should display the corresponding day for the given number. For example, if the user types 1, the output should be Sunday. If the user types 7, the output should be Saturday. If the number is not between 1 to 7 user should get error message as shown in sample output.

7. Write a program that prompts the user to enter their weight (in kilograms) and height (in meters). The program should calculate the Body Mass Index (BMI) using the formula: BMI = weight / (height * height). The program should then classify the BMI into one of the following categories:

```
less than 18.5 - Underweight
BMI between 18.5 and 24.9 - Normal weight
BMI between 25 and 29.9 - Overweight
BMI 30 or greater - Obesity
```

8. The marks obtained by a student in 3 different subjects are input by the user. Your program should calculate the average of subjects and display the grade. The student gets a grade as per the following rules:

Average	Grade
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

10. Write a program that prompts the user to enter three numbers and sorts them in ascending order. The program should print the sorted numbers.

11. Write a program that prompts the user to input three integers and outputs the largest.

12. Write a program that prompts the user to input a character and determine the character is vowel or consonant.

13. Write a program that prompts the user to input a year and determine whether the year is a leap year or not.
Leap Years are any year that can be evenly divided by 4. A year that is evenly divisible by 100 is a leap year only if it is also evenly divisible by 400.

Example :

1992	Leap Year
2000	Leap Year
1900	NOT a Leap Year
1995	NOT a Leap Year

14. Write a program that prompts the user to input number of calls and calculate the monthly telephone bills as per the following rule:
Minimum Rs. 200 for up to 100 calls.
Plus Rs. 0.60 per call for next 50 calls.
Plus Rs. 0.50 per call for next 50 calls.
Plus Rs. 0.40 per call for any call beyond 200 calls.

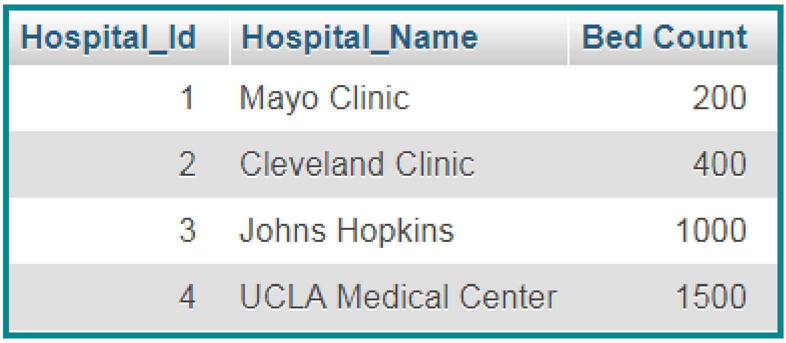
Database questions

Create Database

Create Hospital Table

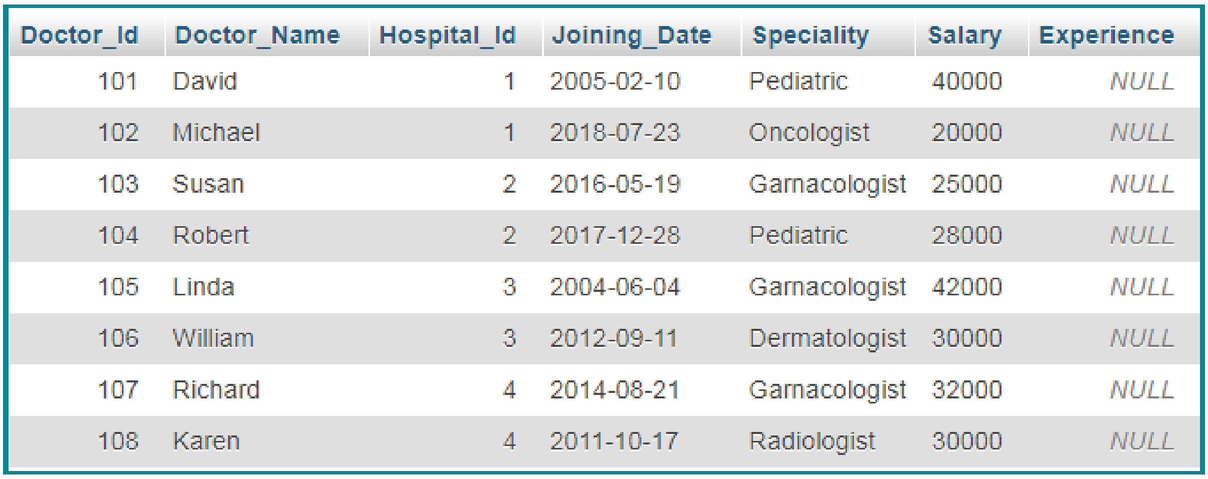
Create Doctor Table

These tables should look like this.



Hospital_Id	Hospital_Name	Bed Count
1	Mayo Clinic	200
2	Cleveland Clinic	400
3	Johns Hopkins	1000
4	UCLA Medical Center	1500

hospital table



Doctor_Id	Doctor_Name	Hospital_Id	Joining_Date	Speciality	Salary	Experience
101	David	1	2005-02-10	Pediatric	40000	*NULL*
102	Michael	1	2018-07-23	Oncologist	20000	*NULL*
103	Susan	2	2016-05-19	Garnacologist	25000	*NULL*
104	Robert	2	2017-12-28	Pediatric	28000	*NULL*
105	Linda	3	2004-06-04	Garnacologist	42000	*NULL*
106	William	3	2012-09-11	Dermatologist	30000	*NULL*
107	Richard	4	2014-08-21	Garnacologist	32000	*NULL*
108	Karen	4	2011-10-17	Radiologist	30000	*NULL*

1: Connect to your database server and print its version

2: Fetch Hospital and Doctor Information using hospital Id and doctor Id

3: Get the list Of doctors as per the given specialty and salary

```
Printing doctors whose specialty is Garnacologist and salary greater than 30000
```

4: Get a list of doctors from a given hospital

Note: Implement the functionality to fetch all the doctors as per the given Hospital Id . **You must display the hospital name of a doctor**.

```
def get_doctors(hospital_id):
    #Fetch All doctors within given Hospital
```

```
get_doctors(2)
```

5:Update doctor experience in years

```
def def update_doctor_experience(doctor_id):
    # Update Doctor Experience in Years
```

```
update_doctor_experience(101)
```