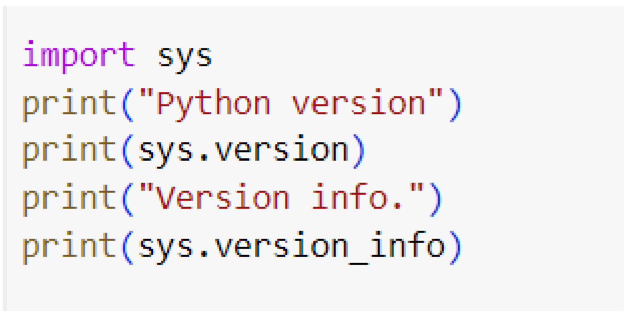
Experiment 1

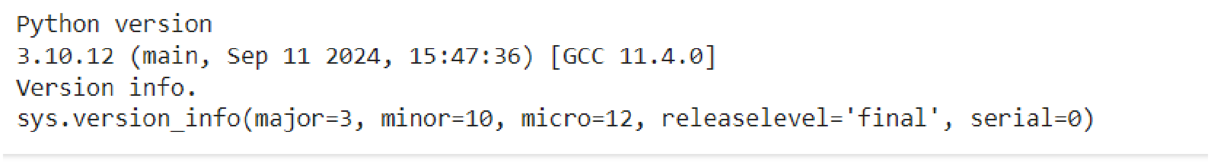
Aim 1: Write a Python program to get the Python version you are using.

Program Code:



```python
import sys
print("Python version")
print(sys.version)
print("Version info.")
print(sys.version_info)
```

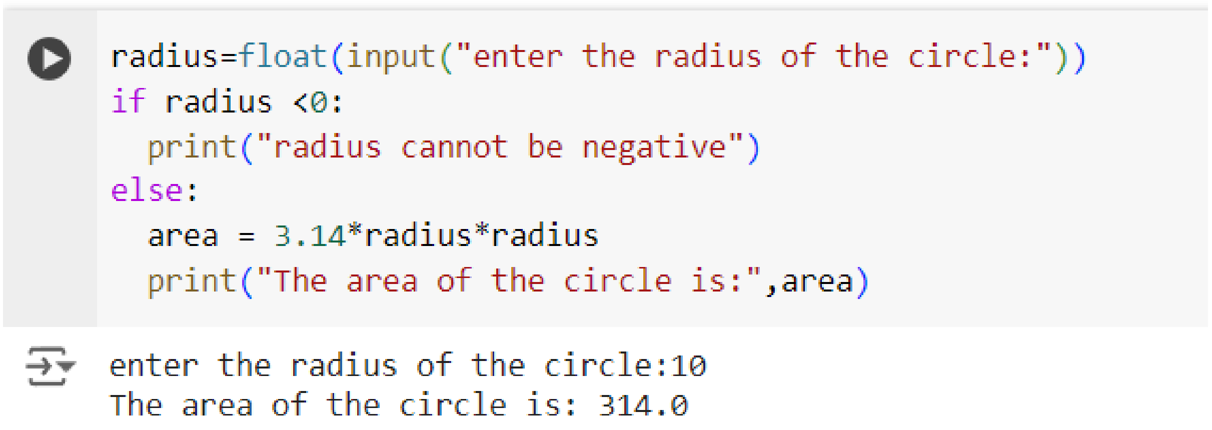
Input and Output:



```
Python version
3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0]
Version info.
sys.version_info(major=3, minor=10, micro=12, releaselevel='final', serial=0)
```

Aim 2: Write a Python program which accepts the radius of a circle from the user and compute the area.

Program Code:



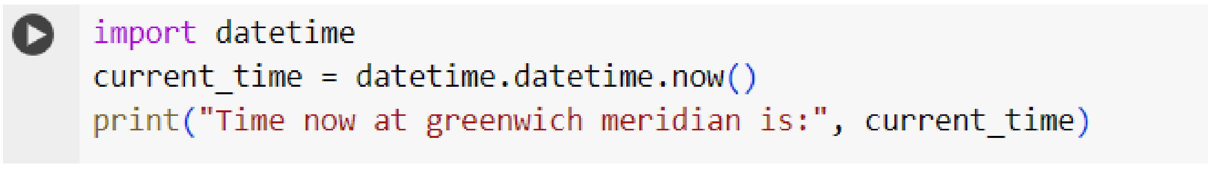
```python
radius=float(input("enter the radius of the circle:"))
if radius <0:
  print("radius cannot be negative")
else:
  area = 3.14*radius*radius
  print("The area of the circle is:",area)
```

```
enter the radius of the circle:10
The area of the circle is: 314.0
```

Experiment 2

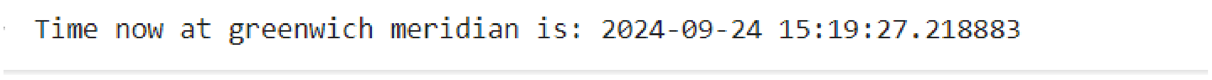
Aim 1: Write a Python program to display the current date and time.

Program Code:



```python
import datetime
current_time = datetime.datetime.now()
print("Time now at greenwich meridian is:", current_time)
```

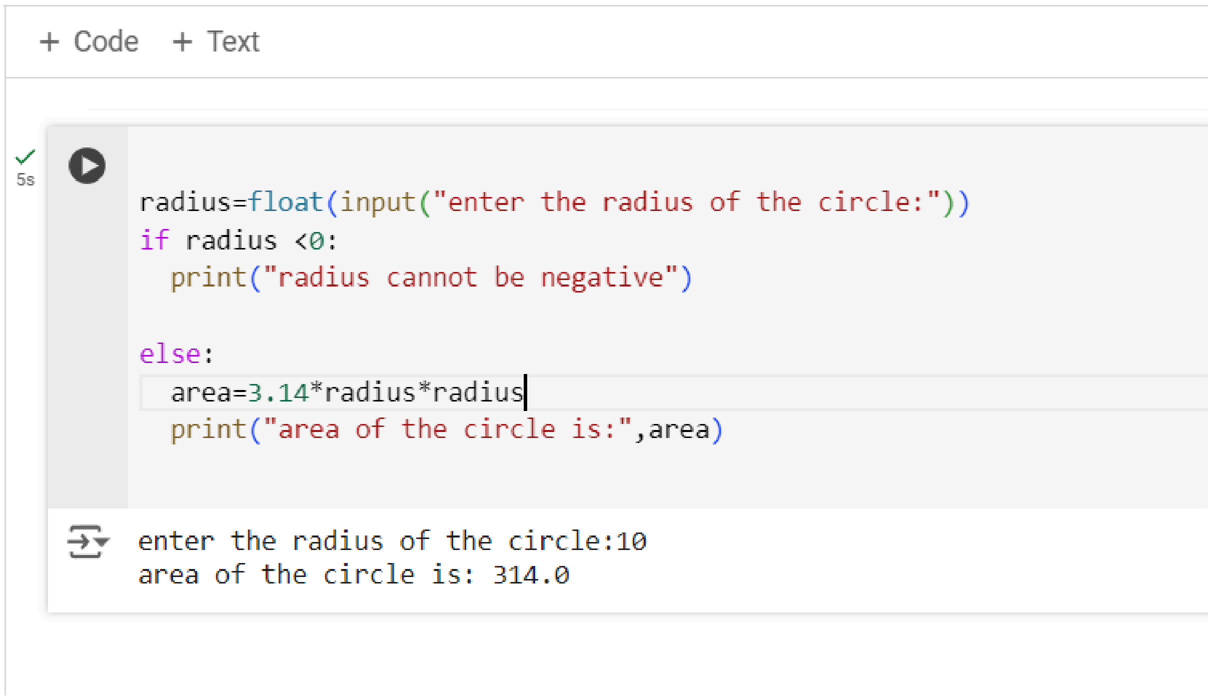
Input and Output:



```
Time now at greenwich meridian is: 2024-09-24 15:19:27.218883
```

Aim 2: Write a Python program which accepts the radius of a circle from the user and compute the area

Program Code:



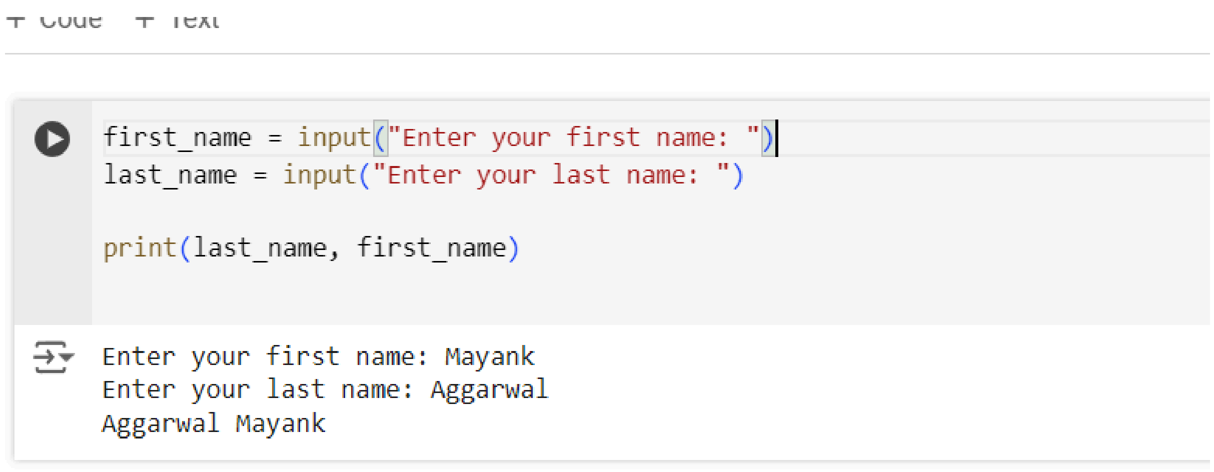
```python
radius=float(input("enter the radius of the circle:"))
if radius <0:
  print("radius cannot be negative")

else:
  area=3.14*radius*radius
  print("area of the circle is:",area)
```

```
enter the radius of the circle:10
area of the circle is: 314.0
```

Experiment 3

Aim 1: Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them.

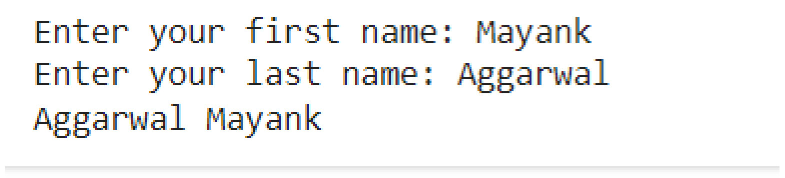
Program Code:



```
first_name = input("Enter your first name: ")
last_name = input("Enter your last name: ")

print(last_name, first_name)
```

```
Enter your first name: Mayank
Enter your last name: Aggarwal
Aggarwal Mayank
```

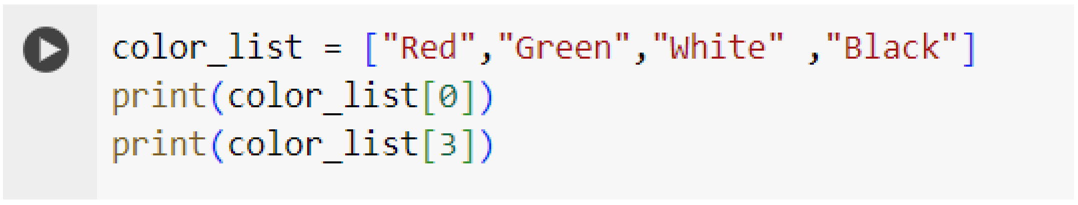
Input and Output:



```
Enter your first name: Mayank
Enter your last name: Aggarwal
Aggarwal Mayank
```

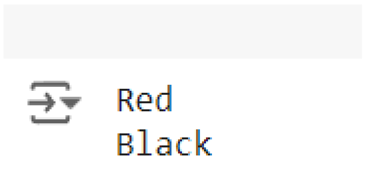
Aim 2: Write a Python program to display the first and last colors from the following list. color_list = ["Red","Green","White" ,"Black"].

Program Code:



```python
color_list = ["Red","Green","White" ,"Black"]
print(color_list[0])
print(color_list[3])
```

Input and Output:



```
Red
Black
```

Experiment 4

Aim: Write a Python program to print the documents (syntax, description etc.) of Python built-in function(s).

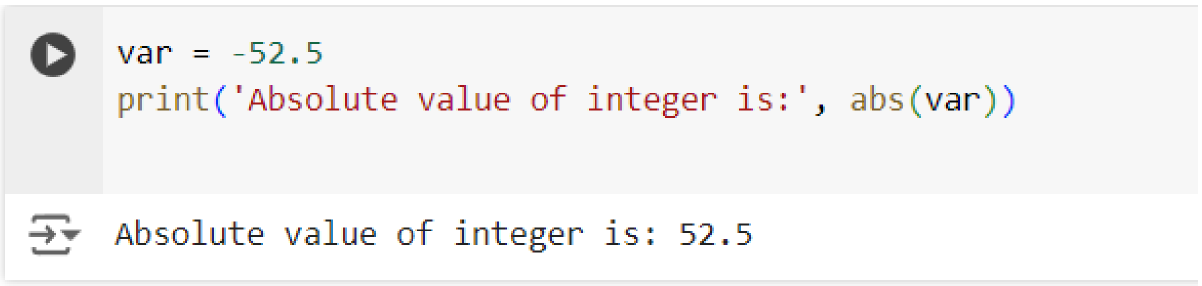
Sample function : abs()

Expected Result :

abs(number) -> number

Return the absolute value of the argument.

Program Code:



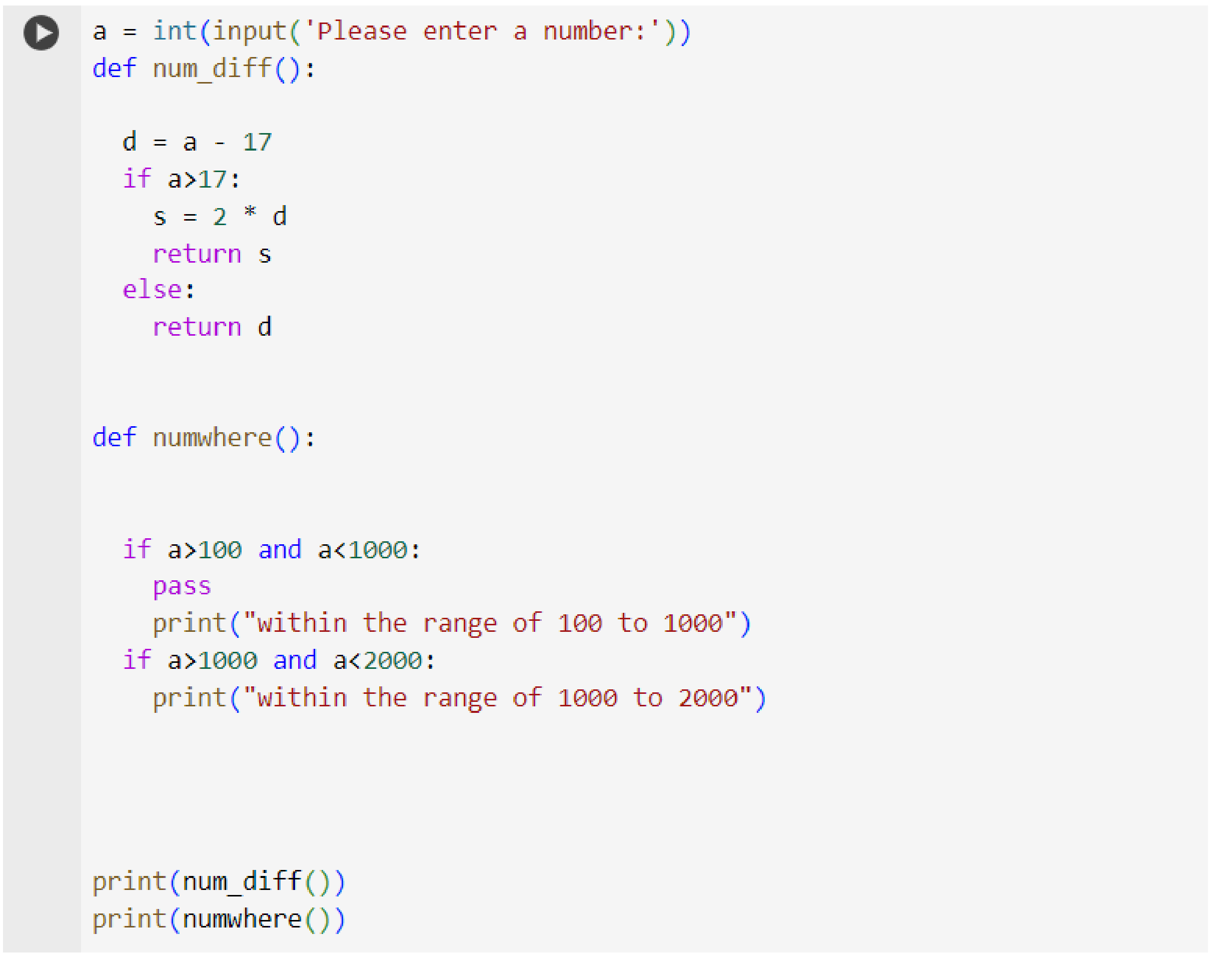
```python
var = -52.5
print('Absolute value of integer is:', abs(var))
```

```
Absolute value of integer is: 52.5
```

Experiment 5

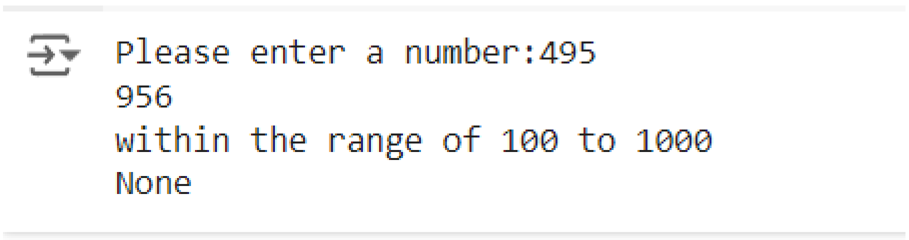
Aim 1: Write a Python program to get the difference between a given number and 17, if the number is greater than 17 return double the absolute difference. Write a Python program to test whether a number is within 100 of 1000 or 2000.

Program Code:

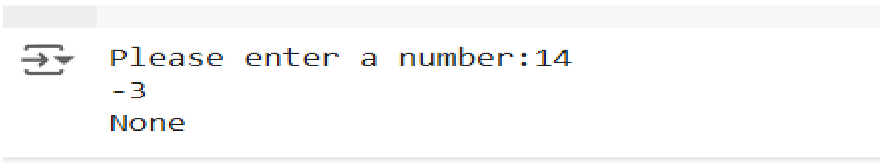


```python
a = int(input('Please enter a number:'))
def num_diff():

  d = a - 17
  if a>17:
    s = 2 * d
    return s
  else:
    return d


def numwhere():


  if a>100 and a<1000:
    pass
    print("within the range of 100 to 1000")
  if a>1000 and a<2000:
    print("within the range of 1000 to 2000")




print(num_diff())
print(numwhere())
```

Input and Output:



```
Please enter a number:495
956
within the range of 100 to 1000
None
```



```
Please enter a number:14
-3
None
```

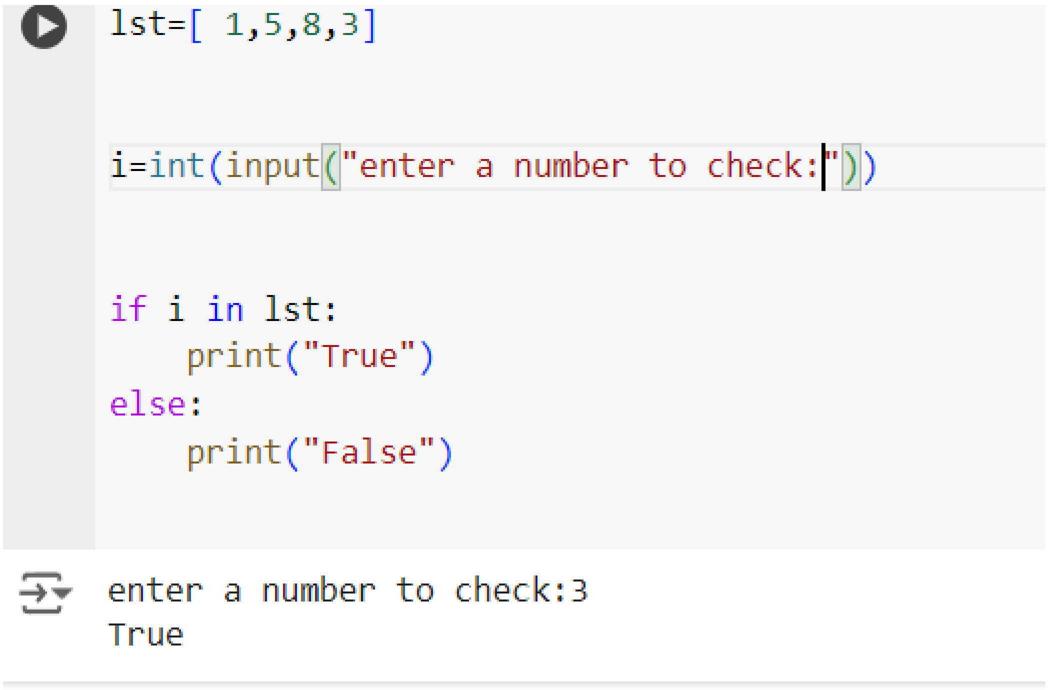
Aim 2: Write a Python program to check whether a specified value is contained in a group of values.

Test Data :

3 -> [1, 5, 8, 3] : True

-1 -> [1, 5, 8, 3] : False

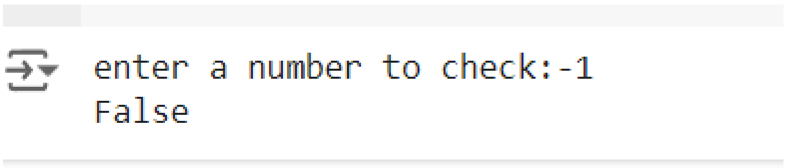
Program Code:



```
lst=[ 1,5,8,3]


i=int(input("enter a number to check:"))


if i in lst:
    print("True")
else:
    print("False")
```

```
enter a number to check:3
True
```

Input and Output:


```
enter a number to check:-1
False
```

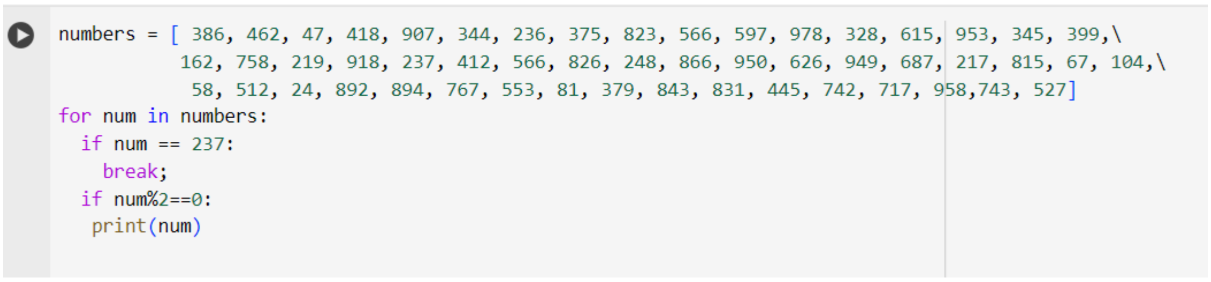
Experiment 6

Aim: Write a Python program to print all even numbers from a given numbers list in the same order and stop the printing if any numbers that come after 237 in the sequence.

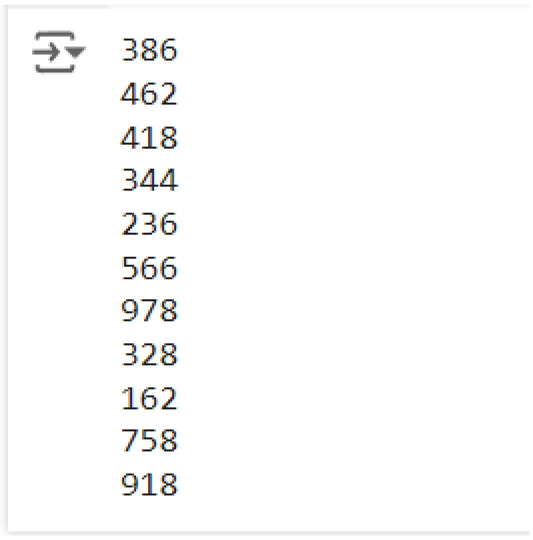
Sample numbers list :

numbers = [386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345, 399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815, 67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717, 958,743, 527]

Program Code:


```
numbers = [ 386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345, 399,\
           162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815, 67, 104,\
            58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717, 958,743, 527]
for num in numbers:
  if num == 237:
    break;
  if num%2==0:
   print(num)
```

Input and Output;



```
386
462
418
344
236
566
978
328
162
758
918
```

Experiment 7

Aim: Write a Python program that accepts a single integer value entered by the user. If the value entered is less than one, the program prints nothing.

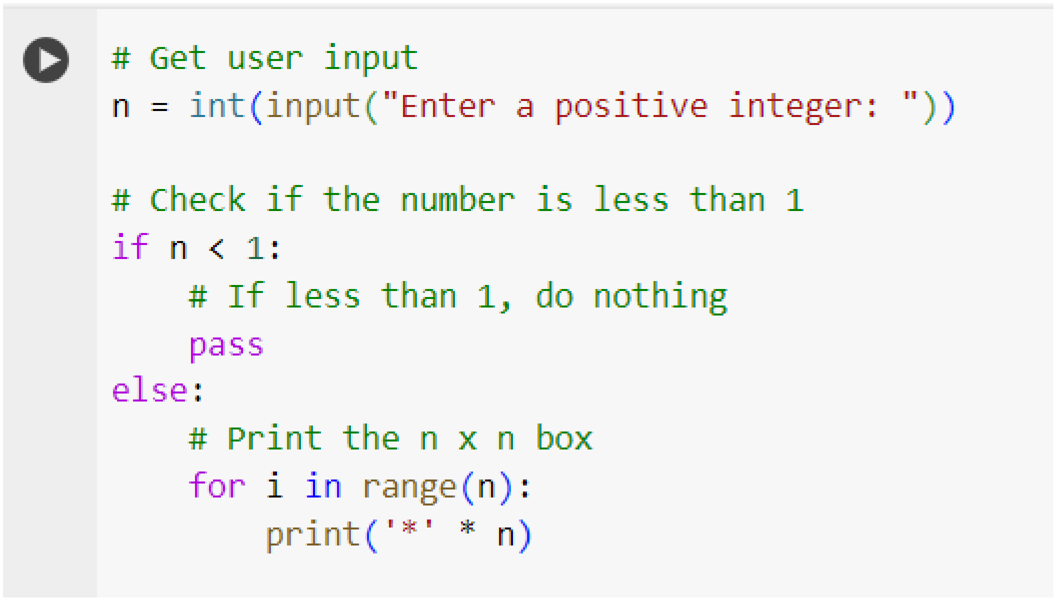
If the user enters a positive integer, n, the program prints an n×n box drawn with * characters.

If the users enters 1, for example, the program prints *

If the user enters a 2, it prints

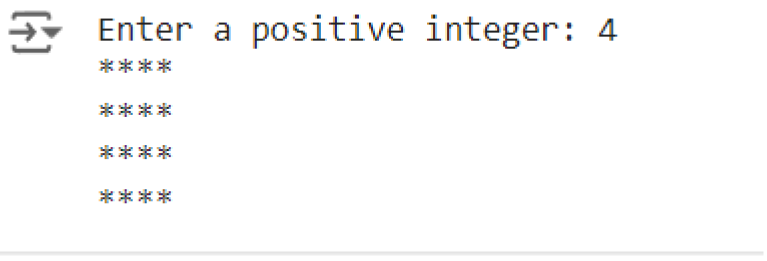
```
**
**
```

Program Code:



```python
# Get user input
n = int(input("Enter a positive integer: "))

# Check if the number is less than 1
if n < 1:
    # If less than 1, do nothing
    pass
else:
    # Print the n x n box
    for i in range(n):
        print('*' * n)
```

Input and Output:

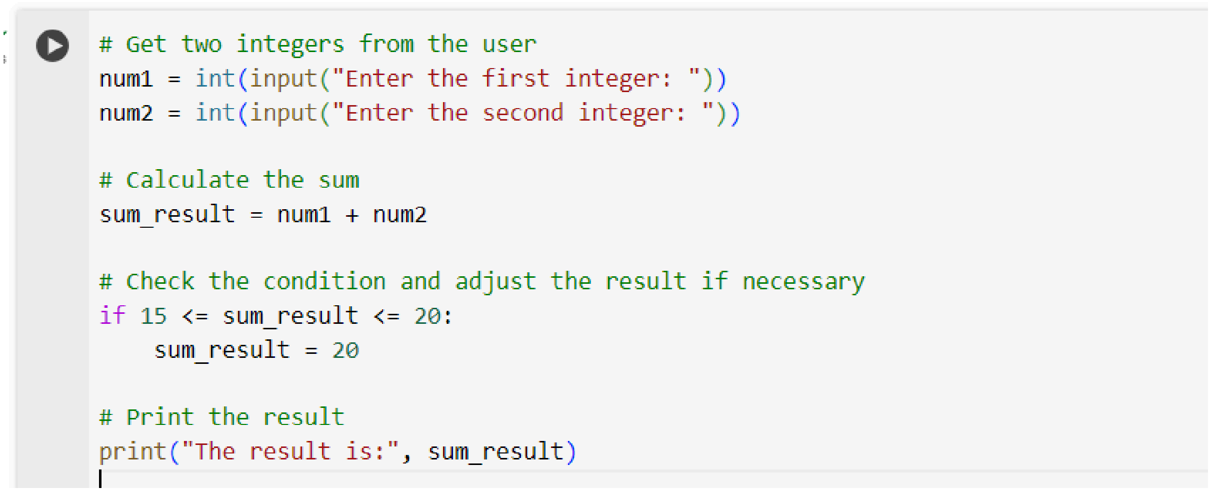


```
Enter a positive integer: 4
****
****
****
****
```

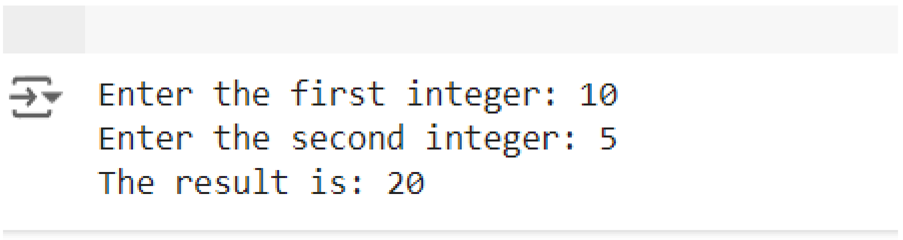
Experiment 8

Aim 1: Write a Python program to add two given integers. However, if the sum is between 15 to 20 it will return 20.

Program Code:


```python
# Get two integers from the user
num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))

# Calculate the sum
sum_result = num1 + num2

# Check the condition and adjust the result if necessary
if 15 <= sum_result <= 20:
    sum_result = 20

# Print the result
print("The result is:", sum_result)
```

Input and Output:

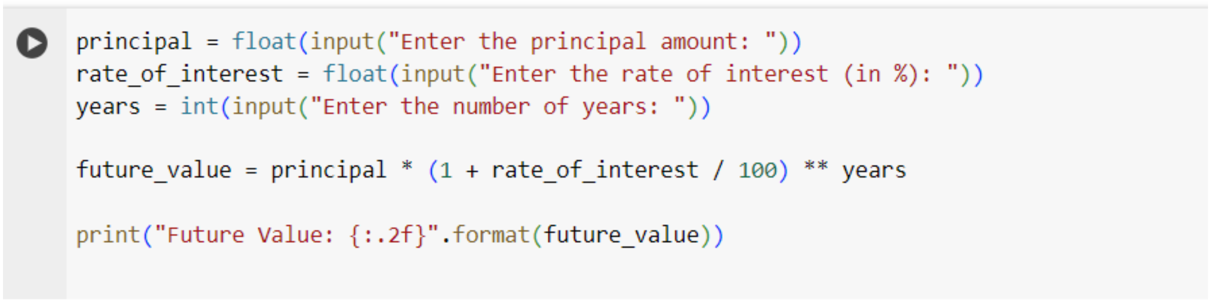

```
Enter the first integer: 10
Enter the second integer: 5
The result is: 20
```

Aim 2: Write a Python program to compute the future value of a specified principal amount, rate of interest, and a number of years.

Test Data : amt = 10000, int = 3.5, years = 7

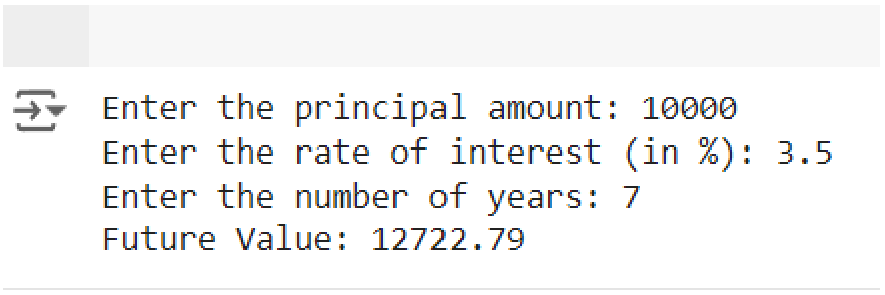
Expected Output : 12722.79

Program Code:



```python
principal = float(input("Enter the principal amount: "))
rate_of_interest = float(input("Enter the rate of interest (in %): "))
years = int(input("Enter the number of years: "))

future_value = principal * (1 + rate_of_interest / 100) ** years

print("Future Value: {:.2f}".format(future_value))
```

Input and Output:



```
Enter the principal amount: 10000
Enter the rate of interest (in %): 3.5
Enter the number of years: 7
Future Value: 12722.79
```

Experiment 10

Task:

Write a Python program to display all the member name of an enum class ordered by their values.
Expected Output:
Country Name ordered by Country Code: Afghanistan Algeria Angola Albania Andorra Antarctica

Sample Solution:

Python Code:

```
from enum import Enum

class Country(Enum):
    AFGHANISTAN = "AF"
    ALGERIA = "DZ"
    ANGOLA = "AO"
    ALBANIA = "AL"
    ANDORRA = "AD"
    ANTARCTICA = "AQ"

def display_countries():
    countries_ordered = sorted(Country, key=lambda c: c.value)
    print("Country Name ordered by Country Code:", end=' ')
    for country in countries_ordered:
        print(country.name.capitalize(), end=' ')
    print()  # To add a newline after the output

# Call the function
display_countries()
```

```
Country Name ordered by Country Code: Andorra Afghanistan Albania Angola
Antarctica Algeria
```

Explanation:

Here, we define an enum class called Country, where each country is a member associated with its corresponding country code (e.g., Afghanistan with "AF"). Each member is unique and has a defined value.

Experiment 12

Task:

Write a Python program to get an array buffer information Expected Output: Array buffer start address in memory and number of elements. (25855056, 2)

Sample Solution:

Python Code:

```
import array

# Create an array
arr = array.array('i', [1, 2])  # 'i' indicates an array of integers

# Get buffer information
buffer_info = arr.buffer_info()

# Print the start address and number of elements
print(buffer_info)
```

Explanation:

The provided Python program uses the array module to create an array of integers, specifically initialized with the values 1 and 2. It then calls the buffer_info() method on the array, which returns a tuple containing the memory address where the array starts and the total number of elements in the array. Finally, the program prints this buffer information, which typically appears in the format (address, number of elements), such as (25855056, 2), demonstrating how to efficiently manage and access homogeneous data types in Python.

Experiment 11

Task:

Write a Python program to get all values from an enum class.
Expected output: [93, 355, 213, 376, 244, 672].

Sample Solution:

Python Code:

```python
from enum import Enum

# Define an Enum class with some values
class Numbers(Enum):
    FIRST = 93
    SECOND = 355
    THIRD = 213
    FOURTH = 376
    FIFTH = 244
    SIXTH = 672
values = [number.value for number in Numbers]
print(values)
```

```
[93, 355, 213, 376, 244, 672]
```

Explanation:

Enum Class: The Numbers enum class is defined with several members and their corresponding integer values. Retrieving Values: A list comprehension is used to iterate through the enum members and collect their values. Output: Finally, it prints the list of values.

Experiment 13, 14

Task:

Write a Python program to push three items into a heap and return the smallest item from the heap. Also Pop and return the smallest item from the heap ExpectedOutput:
Items in the heap:
('V', 1)
('V', 3)
('V', 2)
The smallest item in the heap:
('V', 1)
Pop the smallest item in the heap:
('V', 2) ('V', 3)

Sample Solution:

Python Code:

```
heap = [('v',1),('v',3),('v',2)]

print("Items in the heap:",heap)

Items in the heap: [('v', 1), ('v', 3), ('v', 2)]

def find_smallest_item(heap):
    smallest_item = ('V', float('inf'))
    for item in heap:
        if item[1] < smallest_item[1]:
            smallest_item = item
    return smallest_item

heap=[]
r=int(input("Enter the range of heap:"))
for i in range(r):
  value = int(input(f"Enter value {i + 1} for 'V': "))
  heap.append(('V', value))
print("Items in the heap:")
print(heap)

smallest_item = find_smallest_item(heap)
print("The smallest item in the heap:")
print(smallest_item)

heap.remove(smallest_item)
print("popped item is:",smallest_item)

print("Remaining items in the heap:")
print(heap)
```

```
Enter the range of heap:5
Enter value 1 for 'V': 20
Enter value 2 for 'V': 15
Enter value 3 for 'V': 06
Enter value 4 for 'V': 0
Enter value 5 for 'V': 78
Items in the heap:
[('V', 20), ('V', 15), ('V', 6), ('V', 0), ('V', 78)]
The smallest item in the heap:
('V', 0)
popped item is: ('V', 0)
Remaining items in the heap:
[('V', 20), ('V', 15), ('V', 6), ('V', 78)]
```

Explanation:

In the above code at first we have made an empty heap.then range of heap is taken from the user then after that values of heap has been taken from the user and the values are then inserted into heap using append function.After that the values in the heap are printed and then to find the smallest number in heap smallest_item function is defined.In the function at first value of smallest function is defined as infinty and then it is compared to every element in the heap and is replaced by any elemnt smaller than it.After finding the smallest item in the heap the smallest element is removed from the heap using heap.pop function. After popping the smallest element form the heap it is printed and the remaing heap is printed.

Experiment 15, 16

Task:

Write a function named print_big_enough that accepts two parameters, a list of numbers and a number. The function should print, in order, all the elements in the list that are at least as large as the second parameter.

Sample Solution:

Python Code:

```
def print_big_enough():
    numbers = list(map(int, input("Enter a list of numbers (separated by
spaces): ").split()))
    threshold = int(input("Enter the threshold number: "))

    for number in numbers:
        if number >= threshold:
            print(number)

# Call the function
print_big_enough()
```

```
Enter a list of numbers (separated by spaces): 10 5 6
Enter the threshold number: 5
10
5
6
```

Explanation:

This function allows users to input a list of numbers and a threshold, then prints each number from the list that meets or exceeds the threshold. It effectively combines user input, list processing, and conditional checking in a straightforward manner.