

Qno. 1 Programs of string and operators

The image displays three screenshots of a Jupyter Notebook interface, each showing a different Python program. The interface includes a menu bar (File, Edit, Selection, View, Go, Run), a toolbar with icons for file operations, and a status bar at the bottom showing the current file path and kernel status.

Program 1: Identity Operator

```
# Identity operator
x = [1, 2]
y = x
z = [1, 2]
print(x is y)
print(x is z)
print(x is not z)
```

Output: True, False, True

Program 2: Membership Operator

```
# membership operator
text = "hello"
print("h" in text)
print("z" not in text)
```

Output: True

Program 3: Conditional Operators

```
# if - else
x = 10
if x > 5:
    print("x is greater than 5")
else:
    print("x is 5 or less")

# elif
x = 0
if x > 0:
    print("Positive number")
elif x == 0:
    print("Zero")
else:
    print("Negative number")
```

Output: x is greater than 5, Zero

File Edit Selection View Go Run ...

Select kernel for 'C:\Users\admin\Downloads\TASK 1 string 2.ipynb'

python.exe IntroductiontoMatplotlib (2) Python 3.13.3 --AppData\Local\Microsoft\WindowsApps\python3.13.exe

C:\Users> admin> Downloads> TASK 1 string 2\ Select Another Kernel...

Generate + Code + Markdown | Stop Execution | Clear All Outputs | Go To | Outline ...

Select Kernel

```
s = " hello "
print(s.strip())
```

[4] Python

... hello

```
s = "hello world"
print(s.replace("world", "Python"))
```

[5] Python

... hello Python

```
s = "a,b,c"
print(s.split(","))
```

[6] Python

... ['a', 'b', 'c']

```
words = ["Python", "is", "fun"]
print(" ".join(words))
```

[7] Python

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

PS C:\Users\admin> & C:\Users\admin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/admin/Downloads/import matplotlib.py"

Spaces: 4 | 23:41 01-08-2025

File Edit Selection View Go Run ...

Select kernel for 'C:\Users\admin\Downloads\TASK 1 string 2.ipynb'

python.exe IntroductiontoMatplotlib (2) Python 3.13.3 --AppData\Local\Microsoft\WindowsApps\python3.13.exe

C:\Users> admin> Downloads> TASK 1 string 2\ Select Another Kernel...

Generate + Code + Markdown | Stop Execution | Clear All Outputs | Go To | Outline ...

Select Kernel

```
1
1000
3
```

```
#Relational operators
a = 5
b = 10
print(a == b)
print(a != b)
print(a > b)
print(a < b)
print(a >= b)
print(a <= b)
```

[12] Python

... False
True
False
True
False
True

```
#Assignment operators
v = 5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

PS C:\Users\admin> & C:\Users\admin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/admin/Downloads/import matplotlib.py"

Spaces: 4 | 23:41 01-08-2025

This screenshot shows a Jupyter Notebook with four code cells. The first cell strips whitespace from the string 'hello'. The second cell replaces 'world' with 'Python' in 'hello world'. The third cell splits the string 'a,b,c' by commas. The fourth cell joins the list ['Python', 'is', 'fun'] with spaces. The terminal at the bottom shows the command to run the notebook using Python 3.13.3.

```
s = " hello "
print(s.strip())

s = "hello world"
print(s.replace("world", "Python"))

s = "a,b,c"
print(s.split(","))

words = ["Python", "is", "fun"]
print(" ".join(words))
```

PS C:\Users\admin> & C:\Users\admin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/admin/Downloads/import_matplotlib.py"

This screenshot shows a Jupyter Notebook with three code cells. The first cell prints the length of the string 'Hello'. The second cell prints the lowercase version of 'HELLO'. The third cell prints the uppercase version of 'hello'. The terminal at the bottom shows the command to run the notebook using Python 3.13.3.

```
s = "Hello"
print(len(s))

s = "HELLO"
print(s.lower())

s = "hello"
print(s.upper())

s = " hello "
print(s.strip())
```

PS C:\Users\admin> & C:\Users\admin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/admin/Downloads/import_matplotlib.py"

This screenshot shows a Jupyter Notebook with four code cells, identical to the first screenshot. The first cell strips whitespace from the string 'hello'. The second cell replaces 'world' with 'Python' in 'hello world'. The third cell splits the string 'a,b,c' by commas. The fourth cell joins the list ['Python', 'is', 'fun'] with spaces. The terminal at the bottom shows the command to run the notebook using Python 3.13.3.

```
s = " hello "
print(s.strip())

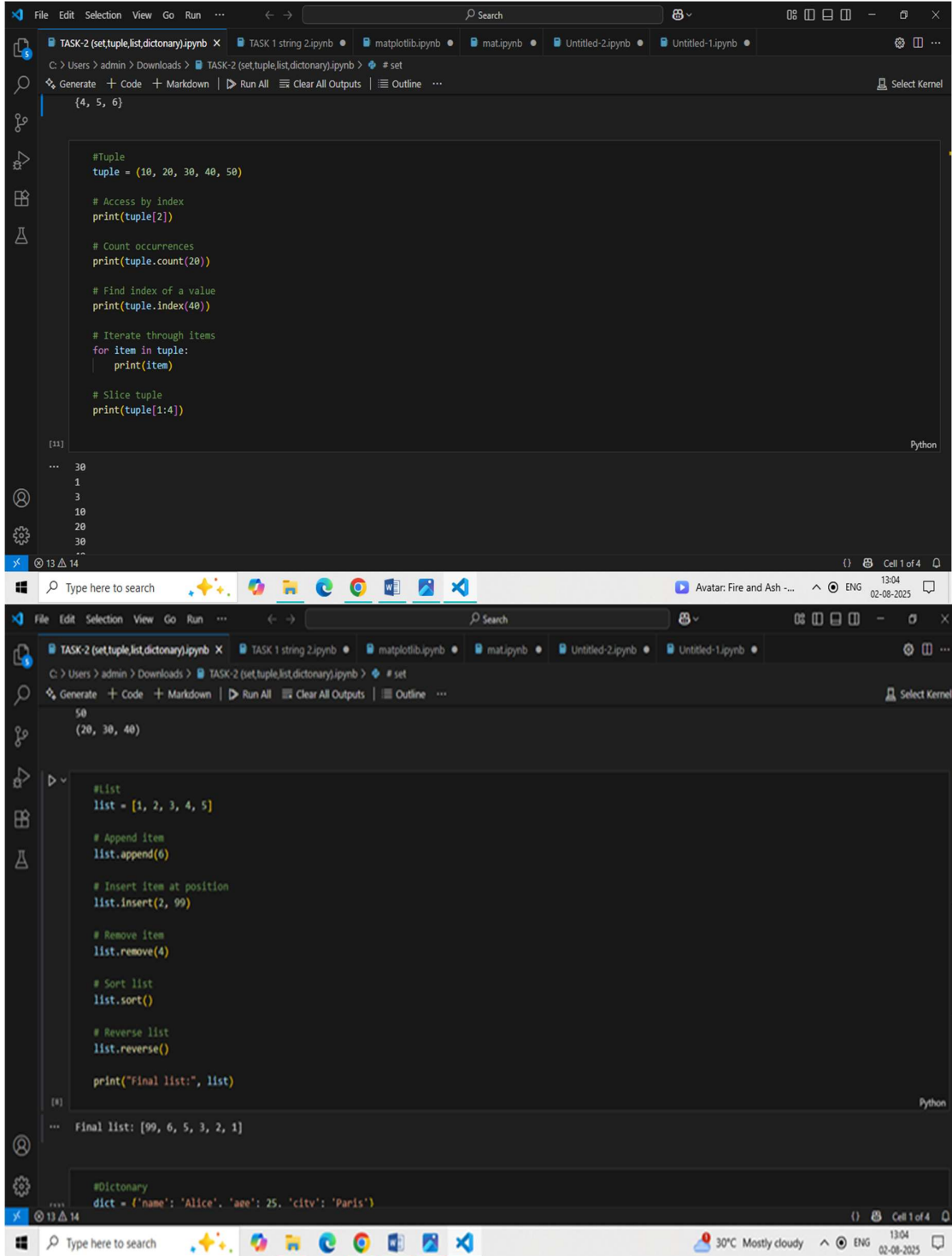
s = "hello world"
print(s.replace("world", "Python"))

s = "a,b,c"
print(s.split(","))

words = ["Python", "is", "fun"]
print(" ".join(words))
```

PS C:\Users\admin> & C:\Users\admin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/admin/Downloads/import_matplotlib.py"

Qno.2 Sets, tuple, list , dictionary



The image displays two screenshots of a Jupyter Notebook interface, showing Python code for working with tuples, lists, and dictionaries.

Top Screenshot: Tuple Operations

```
#Tuple
tuple = (10, 20, 30, 40, 50)

# Access by index
print(tuple[2])

# Count occurrences
print(tuple.count(20))

# Find index of a value
print(tuple.index(40))

# Iterate through items
for item in tuple:
    print(item)

# Slice tuple
print(tuple[1:4])
```

Output:

```
... 30
1
3
10
20
30
--
```

Bottom Screenshot: List and Dictionary Operations

```
#List
list = [1, 2, 3, 4, 5]

# Append item
list.append(6)

# Insert item at position
list.insert(2, 99)

# Remove item
list.remove(4)

# Sort list
list.sort()

# Reverse list
list.reverse()

print("Final list:", list)
```

Output:

```
... Final list: [99, 6, 5, 3, 2, 1]
```

Dictionary

```
dict = {'name': 'Alice', 'age': 25, 'city': 'Paris'}
```

```
File Edit Selection View Go Run ... Search
TASK-2 (set,tuple,list,dictionary).ipynb TASK-1 string 2.ipynb matplotlib.ipynb matplotlib.ipynb Untitled-2.ipynb Untitled-1.ipynb
C:\Users> admin > Downloads > TASK-2 (set,tuple,list,dictionary).ipynb > # set
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ... Select Kernel

# set
set = {1, 2, 3, 4, 5}

# Add an item
set.add(6)

# Remove an item
set.remove(2)

# Check membership
print(3 in set)

# Union with another set
print(set.union({7, 8}))

# Intersection
print(my_set.intersection({4, 5, 6, 10}))

[9] Python

... True
{1, 3, 4, 5, 6, 7, 8}
{4, 5, 6}

# Tuple
tuple = (10, 20, 30, 40, 50)

[11] # Access by index
```

```
File Edit Selection View Go Run ... Search
TASK-2 (set,tuple,list,dictionary).ipynb TASK-1 string 2.ipynb matplotlib.ipynb matplotlib.ipynb Untitled-2.ipynb Untitled-1.ipynb
C:\Users> admin > Downloads > TASK-2 (set,tuple,list,dictionary).ipynb > # set
Generate + Code + Markdown | Run All | Clear All Outputs | Outline ... Select Kernel

# Reverse list
list.reverse()

print("Final list:", list)

[8] Python

... Final list: [99, 6, 5, 3, 2, 1]

# Dictionary
dict = {'name': 'Alice', 'age': 25, 'city': 'Paris'}

# Access value by key
print(dict['name'])

# Add or update key
dict['age'] = 26

# Delete key
del dict['city']

# Get all keys/values
print(dict.keys())
print(dict.values())

# Check if key exists
print('age' in dict)

[13] Python
```

Qno.3 numpy programs

The screenshot displays a Jupyter Notebook with three tasks. The interface includes a top menu bar (File, Edit, Selection, View, Go, Run, ...), a search bar, and a toolbar with icons for file operations, search, and execution. The notebook is open to a file named 'TASK-3 numpy.ipynb'.

Task-1 (Cell 1):
import numpy as np

Task-2 (Cell 2):
#Indexing & slicing
arr = np.array([10, 20, 30, 40, 50])
print(arr[1:4])

Task-3 (Cell 3):
#mathematical operations
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])
print(x + y)
print(x * y)
print(x ** 2)

Task-4 (Cell 4):
#Aggregate function
a = np.array([1, 2, 3, 4])
print(a.sum())
print(a.mean())
print(a.max())

Task-5 (Cell 5):
#Indexing & slicing
arr = np.array([10, 20, 30, 40, 50])
print(arr[1:4])


```
File Edit Selection View Go Run ... Search
TASK-2 (set,tuple,list,dictionary).ipynb TASK-3 numpy.ipynb x TASK 1 string 2.ipynb matplotlib.ipynb mat.ipynb Untitled-2.ipynb Untitled-1.ipynb Select Kernel
C:\Users\> admin > Downloads > TASK-3 numpy.ipynb > import numpy as np
Generate + Code + Markdown Run All Clear All Outputs Outline
[6] #reshape an array
b = np.array([1, 2, 3, 4, 5, 6])
reshaped = b.reshape((2, 3))
Python
[7] #mathematical operations
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])
print(x + y)
print(x * y)
print(x ** 2)
Python
... [5 7 9]
[ 4 10 18]
[1 4 9]
[8] #Aggregate function
a = np.array([1, 2, 3, 4])
print(a.sum())
print(a.mean())
print(a.max())
Python
... 10
13 14
```

```
File Edit Selection View Go Run ... Search
TASK-2 (set,tuple,list,dictionary).ipynb TASK-3 numpy.ipynb x TASK 1 string 2.ipynb matplotlib.ipynb mat.ipynb Untitled-2.ipynb Untitled-1.ipynb Select Kernel
C:\Users\> admin > Downloads > TASK-3 numpy.ipynb > import numpy as np
Generate + Code + Markdown Run All Clear All Outputs Outline
[1] import numpy as np
Python
[2] #Create an array
arr = np.array([1, 2, 3, 4])
Python
[3] # Array of Zeros or Ones
zeros = np.zeros((2, 3))
ones = np.ones((3, 2))
Python
[5] # generate range of values
a = np.arange(0, 10, 2)
Python
[6] #reshape an array
b = np.array([1, 2, 3, 4, 5, 6])
reshaped = b.reshape((2, 3))
Python
13 14
```


Qno.4 Matplotlib programs

