

ICP-1 report



ICP1.ipynb

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

✓ 30s [1] # Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

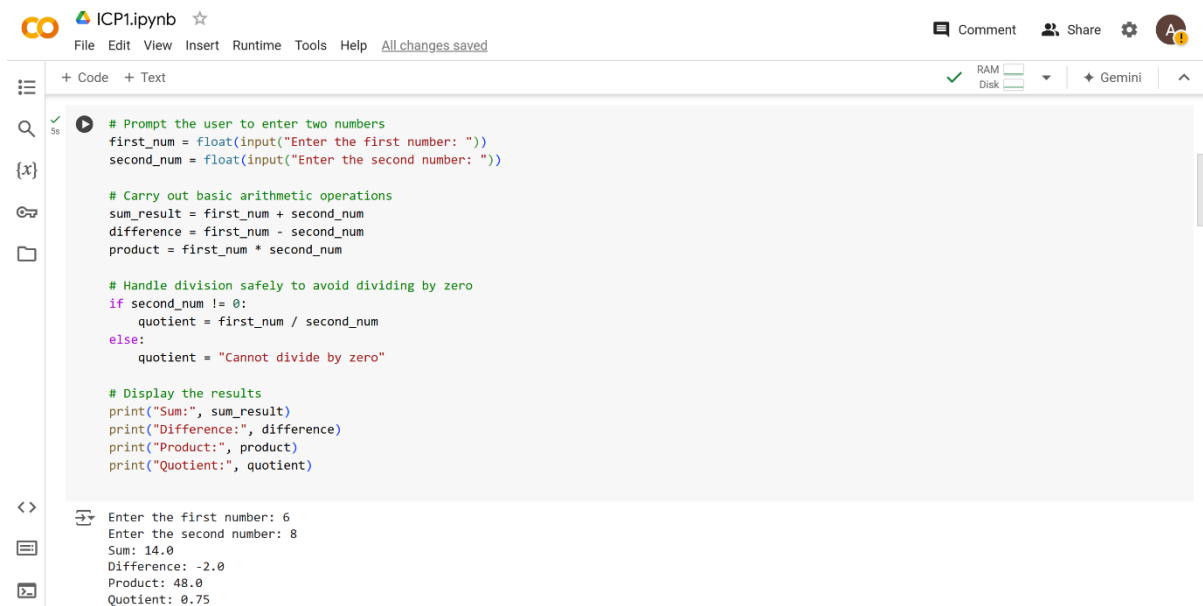
✓ 9s # Get a string from the user
user_input = input("Please enter a string: ")

Remove vowels from the string (a, e, i, o, u)
no_vowels = ''.join([char for char in user_input if char.lower() not in 'aeiou'])

Reverse the modified string
reversed_output = no_vowels[::-1]

Display the final result
print("Processed result:", reversed_output)

Please enter a string: python
Processed result: nhtyp



ICP1.ipynb

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

✓ 5s # Prompt the user to enter two numbers
first_num = float(input("Enter the first number: "))
second_num = float(input("Enter the second number: "))

Carry out basic arithmetic operations
sum_result = first_num + second_num
difference = first_num - second_num
product = first_num * second_num

Handle division safely to avoid dividing by zero
if second_num != 0:
 quotient = first_num / second_num
else:
 quotient = "Cannot divide by zero"

Display the results
print("Sum:", sum_result)
print("Difference:", difference)
print("Product:", product)
print("Quotient:", quotient)

Enter the first number: 6
Enter the second number: 8
Sum: 14.0
Difference: -2.0
Product: 48.0
Quotient: 0.75

ICP1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ RAM
Disk

↓ Gemini ↑

[7] # Prompt the user to enter a sentence
user_text = input("Enter a sentence: ")

Replace each occurrence of 'python' with 'pythons'
modified_text = user_text.replace("python", "pythons")

Display the modified sentence
print("Updated sentence:", modified_text)

Enter a sentence: i love playing with python
Updated sentence: i love playing with pythons

ICP1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ RAM
Disk

↓ Gemini ↑

Insert code cell below (Ctrl+M B)
[8] # Ask the user to enter the class score
score = float(input("Please enter your class score: "))

Determine the grade based on the given score
if score >= 85:
 grade = 'A'
elif 75 <= score < 85:
 grade = 'B'
elif 65 <= score < 75:
 grade = 'C'
elif 55 <= score < 65:
 grade = 'D'
else:
 grade = 'F'

Display the assigned grade
print("Your grade is:", grade)

Please enter your class score: 67
Your grade is: C

ICP1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ RAM
Disk

↓ Gemini ↑

[10] # Define a list with various data types
elements = [23, 'Python', 23.98]

Display the original list
print("Original list:", elements)

Create a new list that contains the types of each element in the original list
type_list = [type(item) for item in elements]

Display the list of types
print("List of types:", type_list)

Original list: [23, 'Python', 23.98]
List of types: [<class 'int'>, <class 'str'>, <class 'float'>]

ICP1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Gemini

05

```
# Initialize sets and list
tech_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
set1 = {19, 22, 24, 20, 25, 26}
set2 = {19, 22, 20, 25, 26, 24, 28, 27}
ages = [22, 19, 24, 25, 26, 24, 25, 24]

# 1. Determine the number of companies in tech_companies
num_tech_companies = len(tech_companies)
print("Number of tech companies:", num_tech_companies)

# 2. Add 'Twitter' to the tech_companies set
tech_companies.add('Twitter')
print("Tech companies after adding Twitter:", tech_companies)

# 3. Add multiple companies to the tech_companies set at once
tech_companies.update(['Snapchat', 'TikTok', 'Spotify'])
print("Tech companies after adding multiple companies:", tech_companies)

# 4. Remove 'Oracle' from the tech_companies set
tech_companies.discard('Oracle') # Using discard to avoid KeyError if 'Oracle' is not present
print("Tech companies after removing a company:", tech_companies)

# 5. Difference between remove and discard:
# - remove() raises a KeyError if the element is not present in the set.
# - discard() does not raise an error if the element is not present.

# 6. Combine set1 and set2
combined_set = set1.union(set2)
```

ICP1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Gemini

05

```
# 6. Combine set1 and set2
combined_set = set1.union(set2)
print("Combined set (set1 union set2):", combined_set)

# 7. Find common elements between set1 and set2
common_elements = set1.intersection(set2)
print("Common elements between set1 and set2:", common_elements)

# 8. Check if set1 is a subset of set2
is_subset = set1.issubset(set2)
print("Is set1 a subset of set2:", is_subset)

# 9. Check if set1 and set2 are disjoint sets
are_disjoint = set1.isdisjoint(set2)
print("Are set1 and set2 disjoint:", are_disjoint)

# 10. Union of set1 with set2 and set2 with set1
set1_union_set2 = set1.union(set2)
set2_union_set1 = set2.union(set1)
print("Set1 union set2:", set1_union_set2)
print("Set2 union set1:", set2_union_set1)

# 11. Find the symmetric difference between set1 and set2
symmetric_diff = set1.symmetric_difference(set2)
print("Symmetric difference between set1 and set2:", symmetric_diff)

# 12. Delete the sets completely
del tech_companies
```



The screenshot shows a Jupyter Notebook titled "ICP1.ipynb". The code cell contains the following Python code:

```
# Add text cell the sets completely
del tech_companies
del set1
del set2

# 13. Convert ages list to a set and compare the length of the list and the set
ages_set = set(ages)
print("Length of ages list:", len(ages))
print("Length of ages set:", len(ages_set))
```

The output of the code is displayed below the code cell:

```
Number of tech companies: 7
Tech companies after adding Twitter: {'Apple', 'Amazon', 'Microsoft', 'Google', 'Facebook', 'Twitter', 'Oracle', 'IBM'}
Tech companies after adding multiple companies: {'Apple', 'Amazon', 'Spotify', 'IBM', 'Google', 'Facebook', 'Twitter', 'Microsoft', 'Oracle', 'Snapch
Tech companies after removing a company: {'Apple', 'Amazon', 'Spotify', 'IBM', 'Google', 'Facebook', 'Twitter', 'Microsoft', 'Snapchat', 'TikTok'}
Combined set (set1 union set2): {19, 20, 22, 24, 25, 26, 27, 28}
Common elements between set1 and set2: {19, 20, 22, 24, 25, 26}
Is set1 a subset of set2: True
Are set1 and set2 disjoint: False
Set1 union set2: {19, 20, 22, 24, 25, 26, 27, 28}
Set2 union set1: {19, 20, 22, 24, 25, 26, 27, 28}
Symmetric difference between set1 and set2: {27, 28}
Length of ages list: 8
Length of ages set: 5
```

My youtube video link: <https://youtu.be/sxEsQud8X4g>

GitHub link: <https://github.com/w8162583/bda.git>