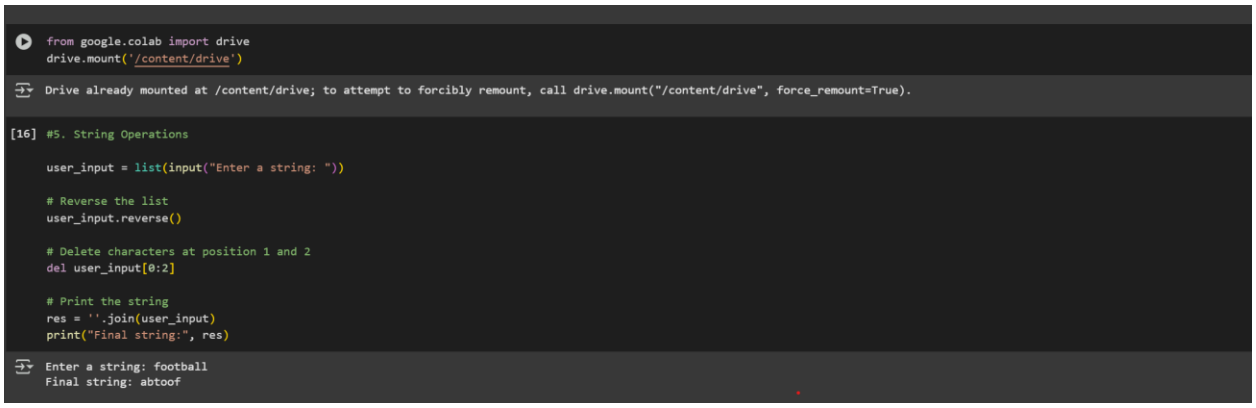
Ganesh Reddy Langati

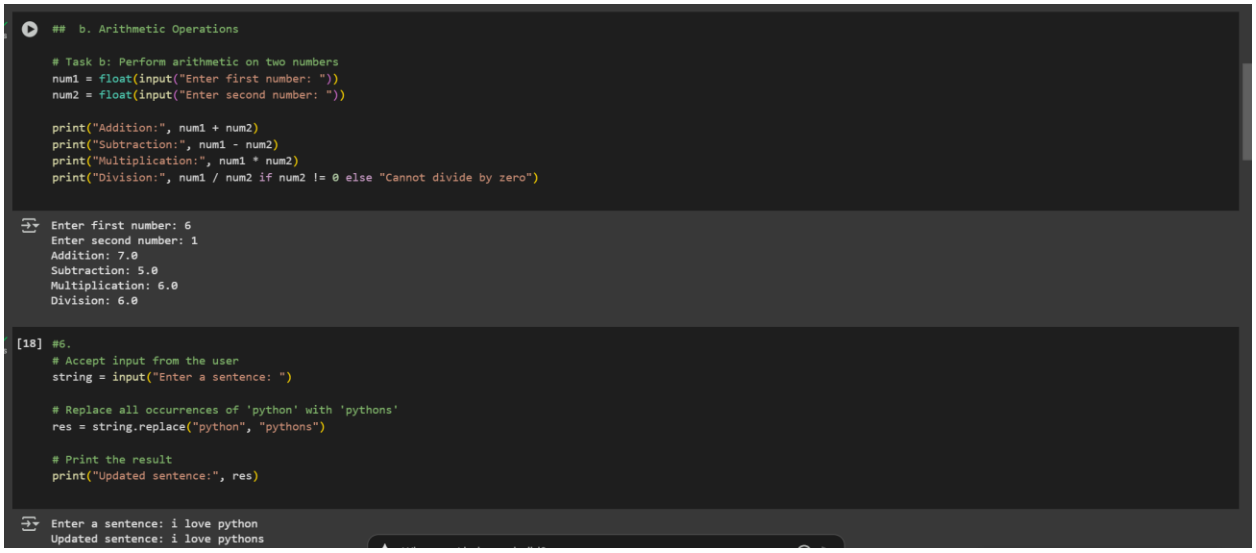


```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[16] #5. String Operations

     user_input = list(input("Enter a string: "))

     # Reverse the list
     user_input.reverse()

     # Delete characters at position 1 and 2
     del user_input[0:2]

     # Print the string
     res = ''.join(user_input)
     print("Final string:", res)
```

```
Enter a string: football
Final string: abtoof
```

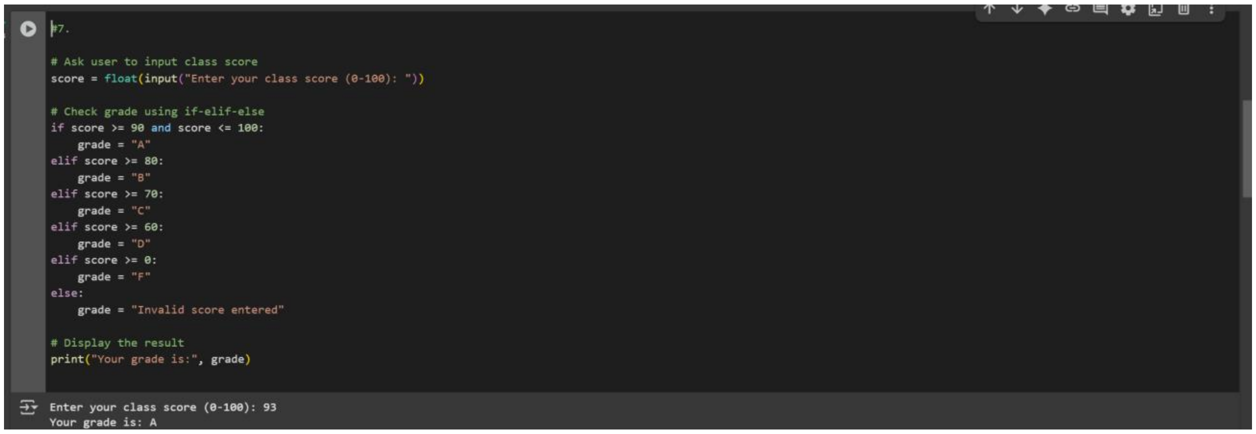


```
## b. Arithmetic Operations

# Task b: Perform arithmetic on two numbers
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

print("Addition:", num1 + num2)
print("Subtraction:", num1 - num2)
print("Multiplication:", num1 * num2)
print("Division:", num1 / num2 if num2 != 0 else "Cannot divide by zero")
```

```
Enter first number: 6
Enter second number: 1
Addition: 7.0
Subtraction: 5.0
Multiplication: 6.0
Division: 6.0
```

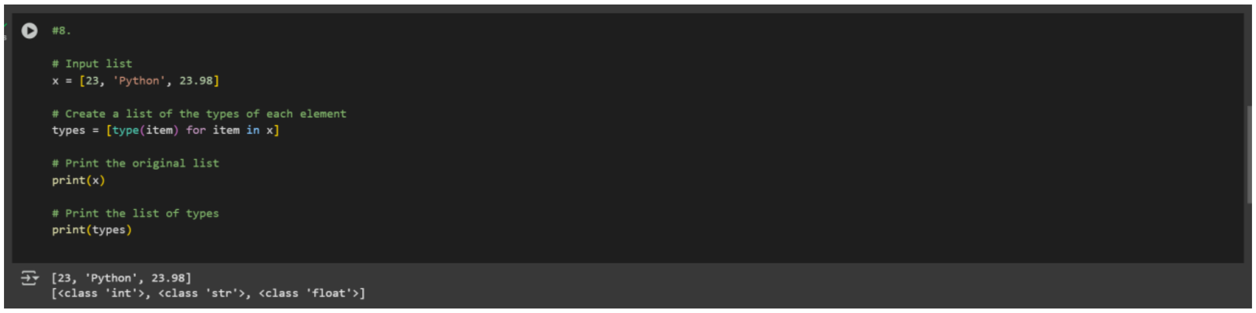
```
[18] #6.
     # Accept input from the user
     string = input("Enter a sentence: ")

     # Replace all occurrences of 'python' with 'pythons'
     res = string.replace("python", "pythons")

     # Print the result
     print("Updated sentence:", res)
```

```
Enter a sentence: i love python
Updated sentence: i love pythons
```



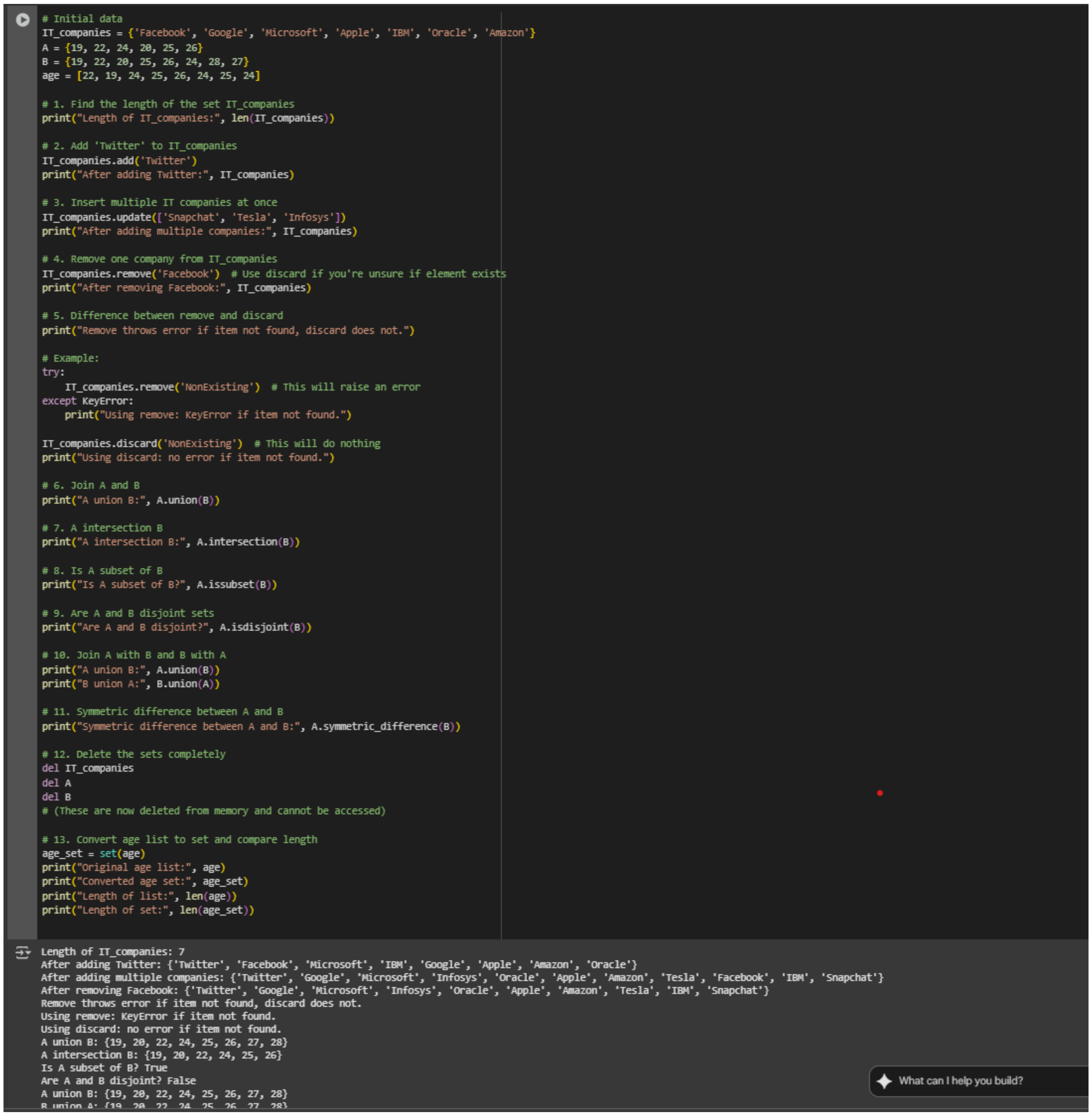
```
#7.

# Ask user to input class score
score = float(input("Enter your class score (0-100): "))

# Check grade using if-elif-else
if score >= 90 and score <= 100:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
elif score >= 0:
    grade = "F"
else:
    grade = "Invalid score entered"

# Display the result
print("Your grade is:", grade)
```

```
Enter your class score (0-100): 93
Your grade is: A
```



```
#8.

# Input list
x = [23, 'Python', 23.98]

# Create a list of the types of each element
types = [type(item) for item in x]

# Print the original list
print(x)

# Print the list of types
print(types)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```



```
# Initial data
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

# 1. Find the length of the set IT_companies
print("Length of IT_companies:", len(IT_companies))

# 2. Add 'Twitter' to IT_companies
IT_companies.add('Twitter')
print("After adding Twitter:", IT_companies)

# 3. Insert multiple IT companies at once
IT_companies.update(['Snapchat', 'Tesla', 'Infosys'])
print("After adding multiple companies:", IT_companies)

# 4. Remove one company from IT_companies
IT_companies.remove('Facebook')  # Use discard if you're unsure if element exists
print("After removing Facebook:", IT_companies)

# 5. Difference between remove and discard
print("Remove throws error if item not found, discard does not.")

# Example:
try:
    IT_companies.remove('NonExisting')  # This will raise an error
except KeyError:
    print("Using remove: KeyError if item not found.")

IT_companies.discard('NonExisting')  # This will do nothing
print("Using discard: no error if item not found.")

# 6. Join A and B
print("A union B:", A.union(B))

# 7. A intersection B
print("A intersection B:", A.intersection(B))

# 8. Is A subset of B
print("Is A subset of B?", A.issubset(B))

# 9. Are A and B disjoint sets
print("Are A and B disjoint?", A.isdisjoint(B))

# 10. Join A with B and B with A
print("A union B:", A.union(B))
print("B union A:", B.union(A))

# 11. Symmetric difference between A and B
print("Symmetric difference between A and B:", A.symmetric_difference(B))

# 12. Delete the sets completely
del IT_companies
del A
del B
# (These are now deleted from memory and cannot be accessed)

# 13. Convert age list to set and compare length
age_set = set(age)
print("Original age list:", age)
print("Converted age set:", age_set)
print("Length of list:", len(age))
print("Length of set:", len(age_set))
```

```
Length of IT_companies: 7
After adding Twitter: {'Twitter', 'Facebook', 'Microsoft', 'IBM', 'Google', 'Apple', 'Amazon', 'Oracle'}
After adding multiple companies: {'Twitter', 'Google', 'Microsoft', 'Infosys', 'Oracle', 'Apple', 'Amazon', 'Tesla', 'Facebook', 'IBM', 'Snapchat'}
After removing Facebook: {'Twitter', 'Google', 'Microsoft', 'Infosys', 'Oracle', 'Apple', 'Amazon', 'Tesla', 'IBM', 'Snapchat'}
Remove throws error if item not found, discard does not.
Using remove: KeyError if item not found.
Using discard: no error if item not found.
A union B: {19, 20, 22, 24, 25, 26, 27, 28}
A intersection B: {19, 20, 22, 24, 25, 26}
Is A subset of B? True
Are A and B disjoint? False
A union B: {19, 20, 22, 24, 25, 26, 27, 28}
B union A: {19, 20, 22, 24, 25, 26, 27, 28}
```