Spring 2025: CS5720

Neural Networks and Deep Learning - ICP-1

Name : Jyothi Kiran Boddeda

Student ID: 700769023

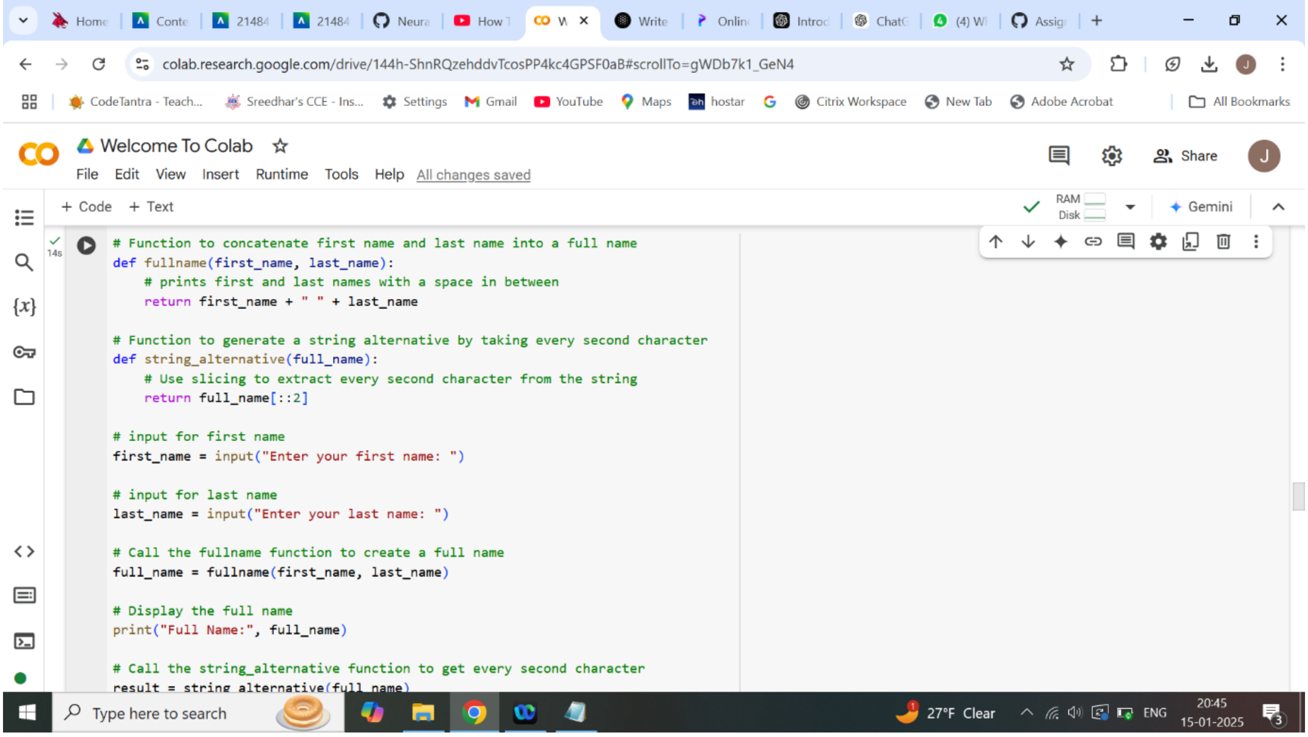
GITHUB LINK : https://github.com/jyothikiranboddeda/Neural-Network-Deep-Learning.git

VIDEO LINK : https://drive.google.com/file/d/1JLLHkrxgEcTF_6GmSOs2GbdGJIceKmtH/view?usp=drive_link

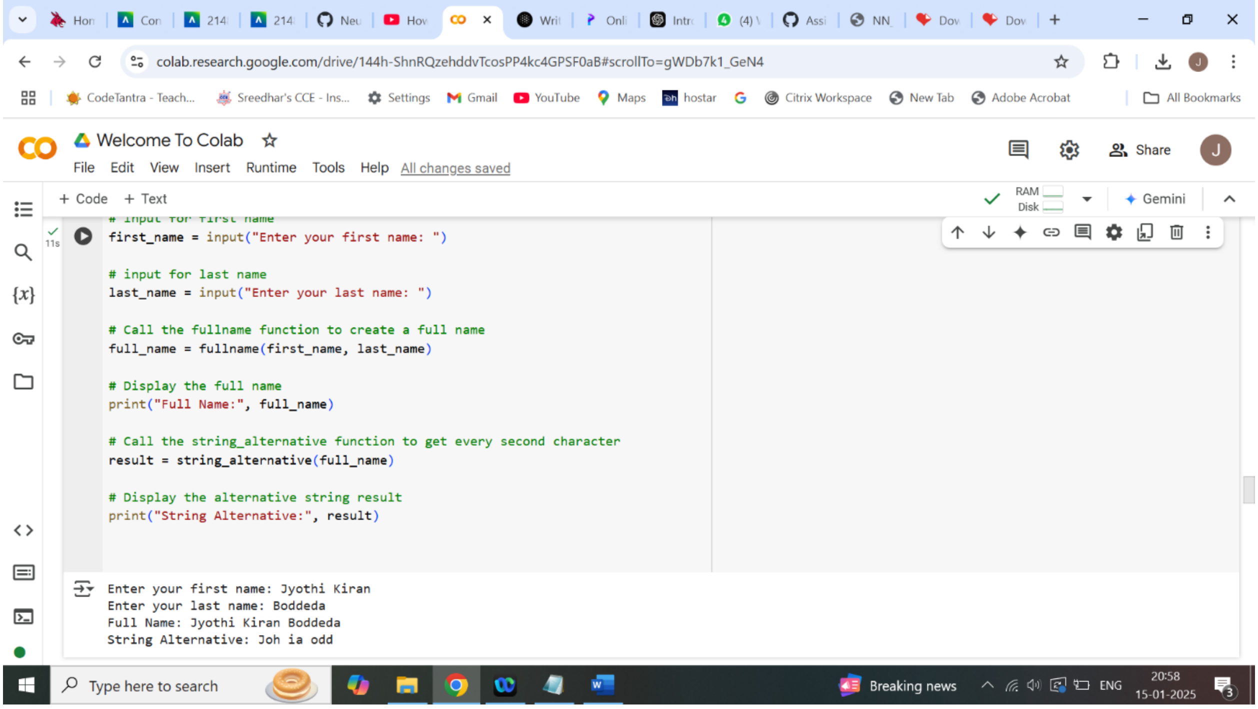
1.Write a program that takes two strings from the user: first_name, last_name. Pass these variables to fullname function that should return the (full name).

- For example:
  - First_name = “your first name”, last_name = “your last name”
  - Full_name = “your full name”
- Write function named “string_alternative” that returns every other char in the full_name string. Str = “**Go**o**d** e**v**e**n**i**n**g”

  Output: Go vnn



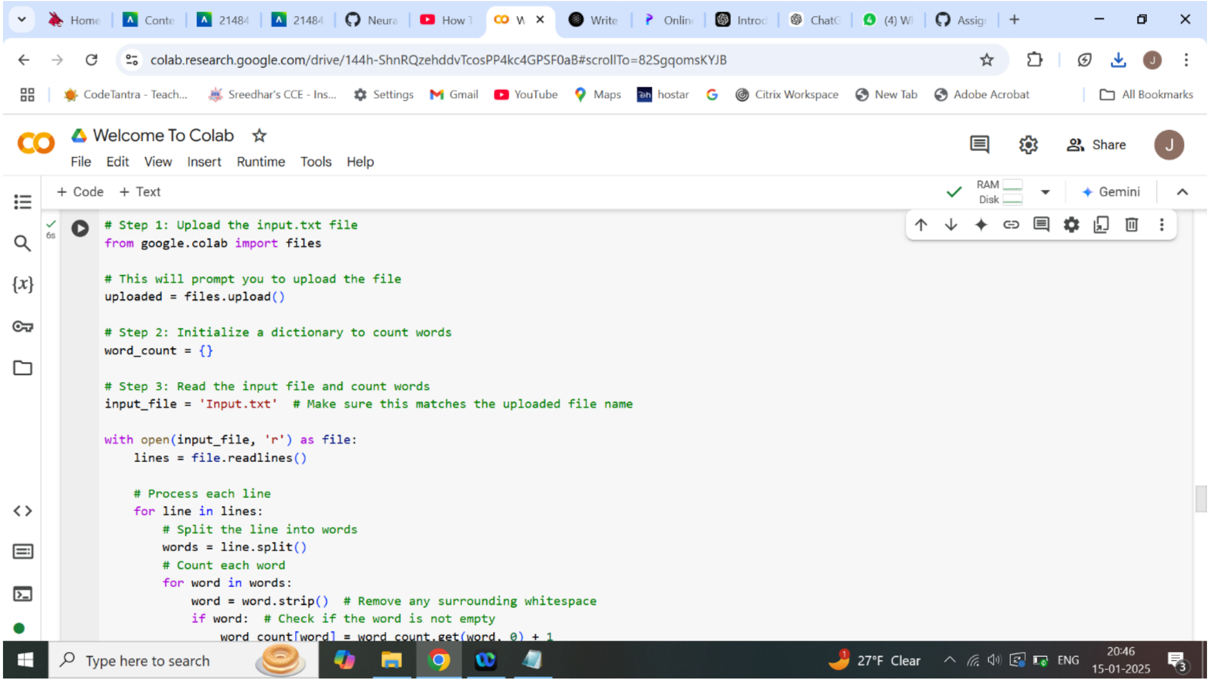
```
# Function to concatenate first name and last name into a full name
def fullname(first_name, last_name):
    # prints first and last names with a space in between
    return first_name + " " + last_name

# Function to generate a string alternative by taking every second character
def string_alternative(full_name):
    # Use slicing to extract every second character from the string
    return full_name[::2]

# input for first name
first_name = input("Enter your first name: ")

# input for last name
last_name = input("Enter your last name: ")

# Call the fullname function to create a full name
full_name = fullname(first_name, last_name)

# Display the full name
print("Full Name:", full_name)

# Call the string_alternative function to get every second character
result = string_alternative(full_name)
```



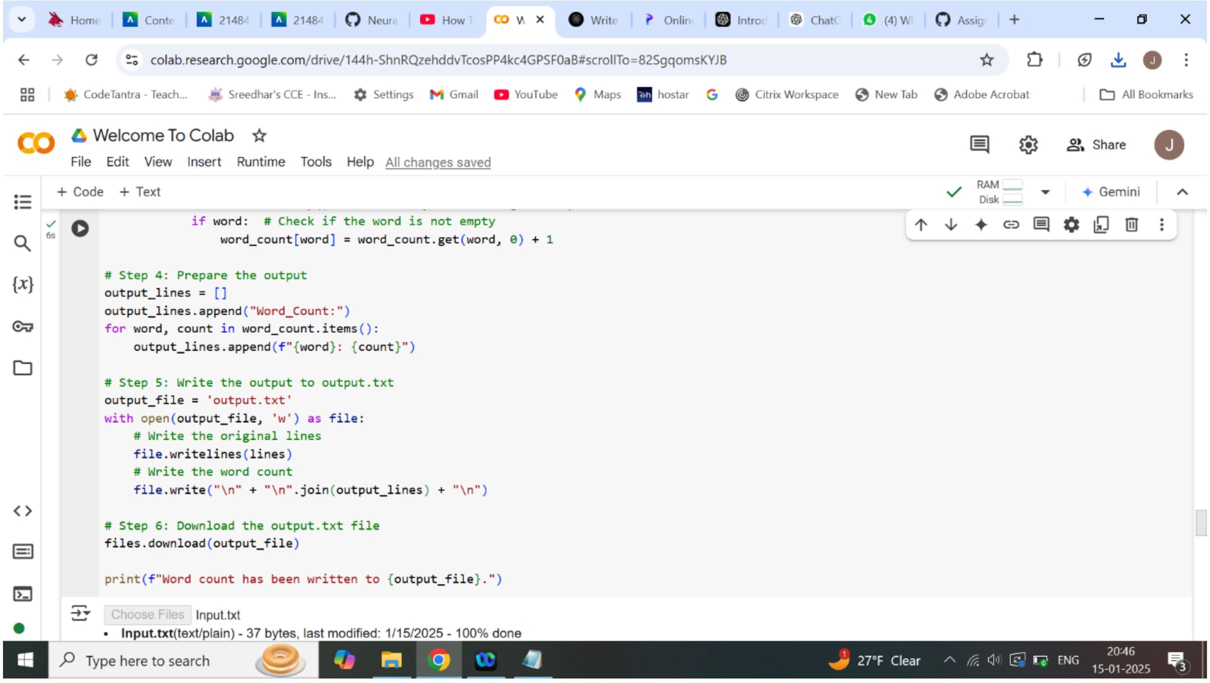
```
# input for first name
first_name = input("Enter your first name: ")

# input for last name
last_name = input("Enter your last name: ")

# Call the fullname function to create a full name
full_name = fullname(first_name, last_name)

# Display the full name
print("Full Name:", full_name)

# Call the string_alternative function to get every second character
result = string_alternative(full_name)

# Display the alternative string result
print("String Alternative:", result)
```

```
Enter your first name: Jyothi Kiran
Enter your last name: Boddeda
Full Name: Jyothi Kiran Boddeda
String Alternative: Joh ia odd
```

2.Write a python program to find the wordcount in a file (input.txt) for each line and then print the output.
Finally store the output in **output.txt** file.

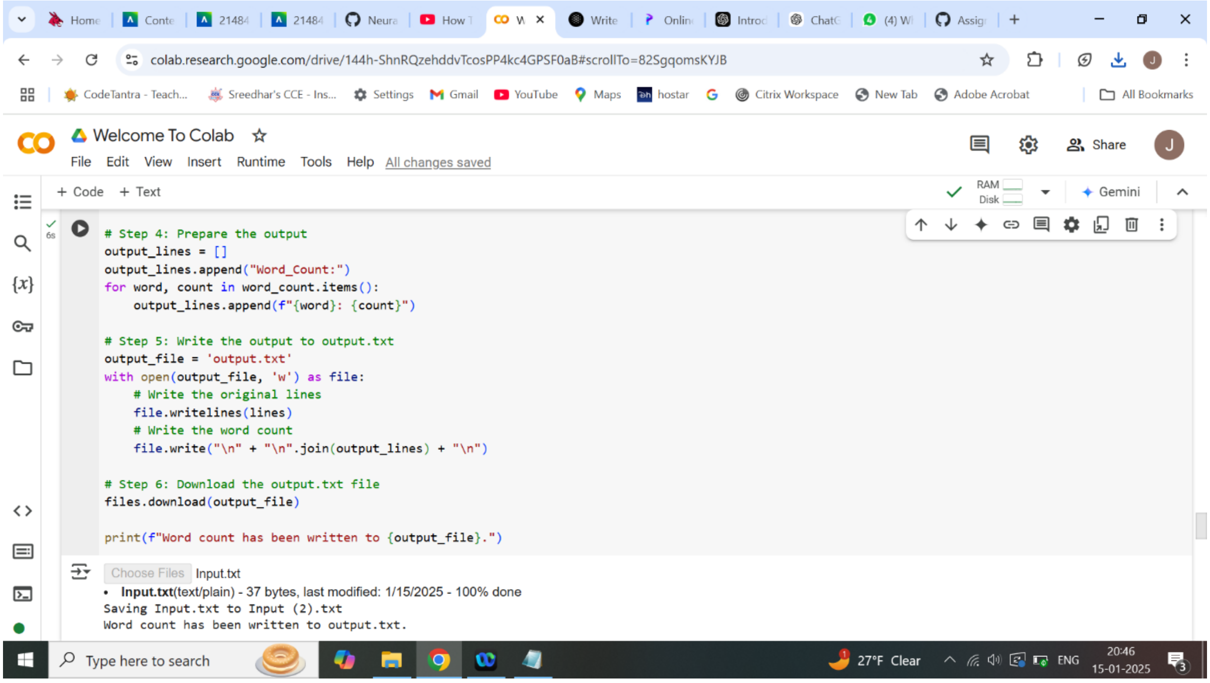


```
# Step 1: Upload the input.txt file
from google.colab import files

# This will prompt you to upload the file
uploaded = files.upload()

# Step 2: Initialize a dictionary to count words
word_count = {}

# Step 3: Read the input file and count words
input_file = 'Input.txt'  # Make sure this matches the uploaded file name

with open(input_file, 'r') as file:
    lines = file.readlines()

    # Process each line
    for line in lines:
        # Split the line into words
        words = line.split()
        # Count each word
        for word in words:
            word = word.strip()  # Remove any surrounding whitespace
            if word:  # Check if the word is not empty
                word_count[word] = word_count.get(word, 0) + 1
```


colab.research.google.com/drive/144h-ShnRQzehddvTcosPP4kc4GPSF0aB#scrollTo=82SgqomsKYJB
Welcome To Colab
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Gemini
Type here to search
27°F Clear
20:46
15-01-2025

```
        if word:  # Check if the word is not empty
            word_count[word] = word_count.get(word, 0) + 1

# Step 4: Prepare the output
output_lines = []
output_lines.append("Word_Count:")
for word, count in word_count.items():
    output_lines.append(f"{word}: {count}")

# Step 5: Write the output to output.txt
output_file = 'output.txt'
with open(output_file, 'w') as file:
    # Write the original lines
    file.writelines(lines)
    # Write the word count
    file.write("\n" + "\n".join(output_lines) + "\n")

# Step 6: Download the output.txt file
files.download(output_file)

print(f"Word count has been written to {output_file}.")
```

```
Choose Files Input.txt
• Input.txt(text/plain) - 37 bytes, last modified: 1/15/2025 - 100% done
```

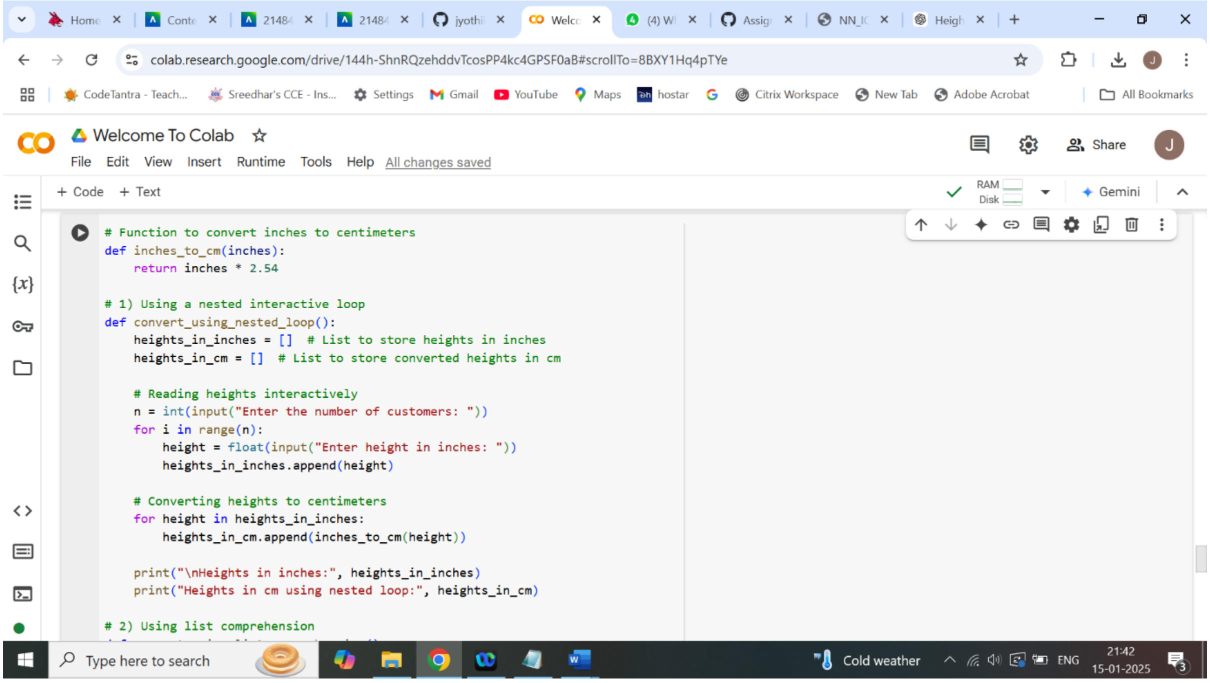

colab.research.google.com/drive/144h-ShnRQzehddvTcosPP4kc4GPSF0aB#scrollTo=82SgqomsKYJB
Welcome To Colab
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Gemini
Type here to search
27°F Clear
20:46
15-01-2025

```
# Step 4: Prepare the output
output_lines = []
output_lines.append("Word_Count:")
for word, count in word_count.items():
    output_lines.append(f"{word}: {count}")

# Step 5: Write the output to output.txt
output_file = 'output.txt'
with open(output_file, 'w') as file:
    # Write the original lines
    file.writelines(lines)
    # Write the word count
    file.write("\n" + "\n".join(output_lines) + "\n")

# Step 6: Download the output.txt file
files.download(output_file)

print(f"Word count has been written to {output_file}.")
```

```
Choose Files Input.txt
• Input.txt(text/plain) - 37 bytes, last modified: 1/15/2025 - 100% done
Saving Input.txt to Input (2).txt
Word count has been written to output.txt.
```

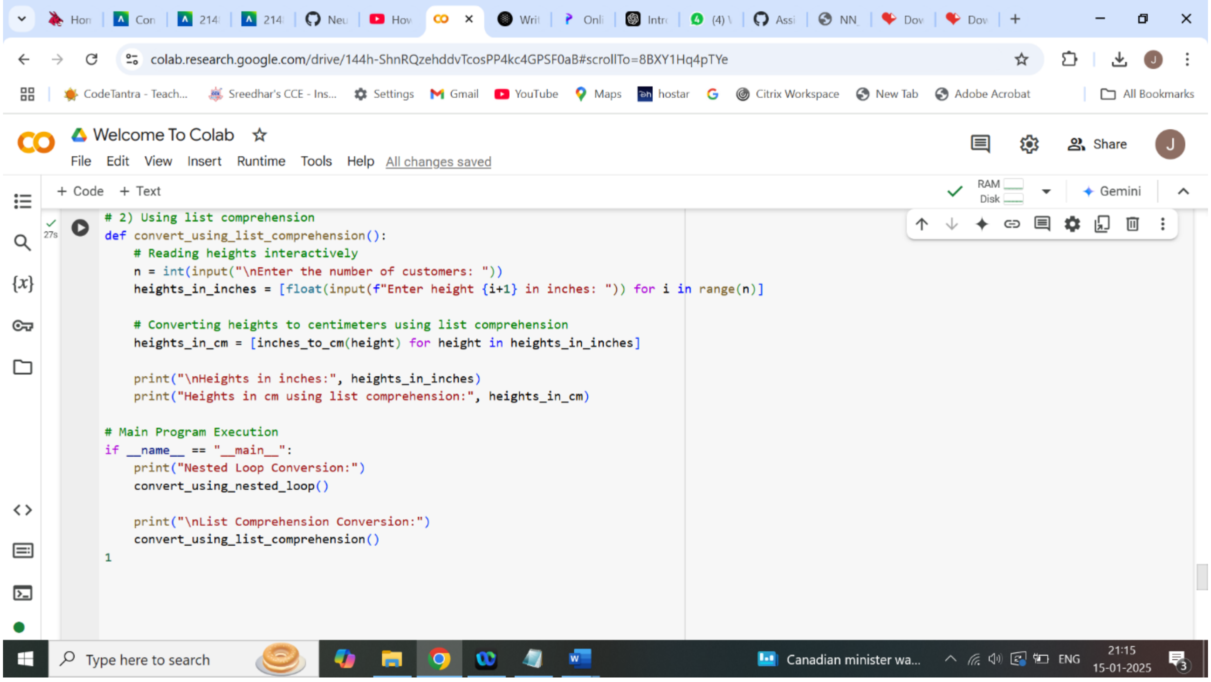
3. Write a program, which reads heights (inches.) of customers into a list and convert these heights to centimeters in a separate list using:

1) Nested Interactive loop.
2) List Comprehensions

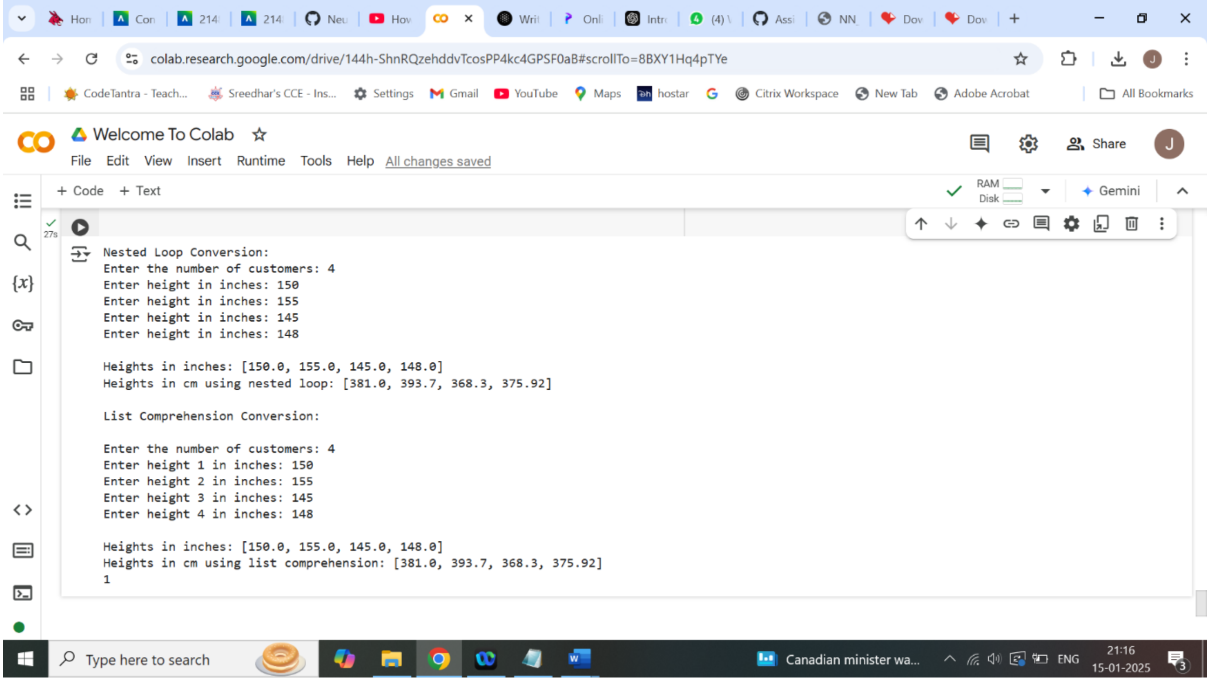


colab.research.google.com/drive/144h-ShnRQzehddvTcosPP4kc4GPSF0aB#scrollTo=8BXY1Hq4pTYe
Welcome To Colab
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
RAM
Disk
Gemini
Share
Type here to search
Cold weather
ENG
21:42
15-01-2025

```python
# Function to convert inches to centimeters
def inches_to_cm(inches):
    return inches * 2.54

# 1) Using a nested interactive loop
def convert_using_nested_loop():
    heights_in_inches = []  # List to store heights in inches
    heights_in_cm = []  # List to store converted heights in cm

    # Reading heights interactively
    n = int(input("Enter the number of customers: "))
    for i in range(n):
        height = float(input("Enter height in inches: "))
        heights_in_inches.append(height)

    # Converting heights to centimeters
    for height in heights_in_inches:
        heights_in_cm.append(inches_to_cm(height))

    print("\nHeights in inches:", heights_in_inches)
    print("Heights in cm using nested loop:", heights_in_cm)

# 2) Using list comprehension
```



```
# 2) Using list comprehension
def convert_using_list_comprehension():
    # Reading heights interactively
    n = int(input("\nEnter the number of customers: "))
    heights_in_inches = [float(input(f"Enter height {i+1} in inches: ")) for i in range(n)]

    # Converting heights to centimeters using list comprehension
    heights_in_cm = [inches_to_cm(height) for height in heights_in_inches]

    print("\nHeights in inches:", heights_in_inches)
    print("Heights in cm using list comprehension:", heights_in_cm)

# Main Program Execution
if __name__ == "__main__":
    print("Nested Loop Conversion:")
    convert_using_nested_loop()

    print("\nList Comprehension Conversion:")
    convert_using_list_comprehension()
1
```



```
Nested Loop Conversion:
Enter the number of customers: 4
Enter height in inches: 150
Enter height in inches: 155
Enter height in inches: 145
Enter height in inches: 148

Heights in inches: [150.0, 155.0, 145.0, 148.0]
Heights in cm using nested loop: [381.0, 393.7, 368.3, 375.92]

List Comprehension Conversion:

Enter the number of customers: 4
Enter height 1 in inches: 150
Enter height 2 in inches: 155
Enter height 3 in inches: 145
Enter height 4 in inches: 148

Heights in inches: [150.0, 155.0, 145.0, 148.0]
Heights in cm using list comprehension: [381.0, 393.7, 368.3, 375.92]
1
```