Code1:

def reverse\_string(s):

    reversed = ""

    for i in range(len(s) - 1, -1, -1):

        reversed += s[i]

    return reversed

def main():

    input\_string = "Hello, world!"

    reversed\_string = reverse\_string(input\_string)

    print(f"Reversed string: {reversed\_string}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

There are no errors in this code it successfully reverses the input string and run the code multiple times I got the output perfectly with the required output and I had compiled successfully

Code2:

There is an error in get\_age function here the corrected code as follows

def get\_age():

age = input("Please enter your age: ")

if age.isnumeric() and int(age) >= 18:

return int(age)

else:

return None

def main():

age = get\_age()

if age:

print(f"You are {age} years old and eligible.")

else:

print("Invalid input. You must be at least 18 years old.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

In this get\_age function we use int(age)>=18 instead of age>=18 because input() returns a string . you need to convert the numeric type that means integer type

With this modification check if it is a valid numeric value and if the age is greater than or equal to 18 then print the result and I can run code after modification I got the result perfectly

Code3:

There are a couple of errors in a given code we need to correct the code as follows

def read\_and\_write\_file(filename):

try:

with open(filename, 'r') as file:

content = file.read()

with open(filename, 'w') as file:

file.write(content.upper())

print(f"File '{filename}' processed successfully.")

except Exception as e:

print(f"An error occurred: {str(e)}")

def main():

filename = "sample.txt"

read\_and\_write\_file(filename)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Error1:the original code first opened the file in ‘r ‘ mode and then in ‘w’ mode without storing the variable it becomes error we need to read the content first and store it in the content variable

Error2:we don’t open the file in ‘ w ‘ mode directly. If we open the file after reading the content then we can write anything in that file

Modify these two corrections then I can run the code and got the output perfectly

Code4:

Here's the corrected code:

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

merge\_sort(left)

merge\_sort(right)

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

j += 1

k += 1

arr = [38, 27, 43, 3, 9, 82, 10]

merge\_sort(arr)

print(f"The sorted array is: {arr}")

The issue in the provided code is that the recursive calls to **merge\_sort(left)** and **merge\_sort(right)** are not updating the original **arr**. In the current implementation, these recursive calls sort the **left** and **right** subarrays but do not merge them back into the original **arr**. To fix this issue, you need to ensure that the merged results from **left** and **right** are correctly placed back into the original **arr**.

With this modification merge sort algorithm should works perfectly I can run the code and I got the output succesfully