Assignment 1

Find and fix the errors in the codes

Code 1

In [5]:
```
def reverse_string(s):    # Changed the variable from reversed which is a built in function to reversed_str
    reversed_str= ""
    for i in range(len(s) - 1, -1, -1):
        reversed_str += s[i]
    return reversed_str

def main():
    input_string = "Hello, world!"
    reversed_string = reverse_string(input_string)
    print(f"Reversed string: {reversed_string}")

if __name__ == "__main__":
    main()
```

```
Reversed string: !dlrow ,olleH
```

Code 2

In [3]:
```
def get_age():
    age = input("Please enter your age: ")
    if age.isnumeric() and age >= str(18): # age and 18 should be of same data type. either
        return int(age)                    #  change both to integer or both to string
    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()
```

```
Please enter your age: 23
You are 23 years old and eligible.
```

Code 3

In [8]:
```
def read_and_write_file(filename):
    try:
        with open("C:\\Users\\hp\\Desktop\\sample.txt", 'r') as file:
            content = file.read()

        # Save the modified content separately before writing
        modified_content = content.upper()

        with open("C:\\Users\\hp\\Desktop\\sample.txt", 'w') as file:
            file.write(modified_content)
        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()
```

```
File 'sample.txt' processed successfully.
```

Code 4

In [10]:
```
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

# Initialize and sort the array
arr = [38, 27, 43, 3, 9, 82, 10]
merge_sort(arr)
print(f"The sorted array is: {arr}")
```

```
The sorted array is: [3, 9, 10, 27, 38, 43, 82]
```

Assignment 2

Grocery Store Inventory management

In [11]:

```
# Initialize an empty inventory dictionary to store items
inventory = {}

def add_item():
    name = input("Enter the name of the item: ")
    quantity = int(input("Enter the quantity of the item: "))
    price = float(input("Enter the price of the item: "))
    inventory[name] = {'quantity': quantity, 'price': price}
    print(f"{name} added to the inventory.")

def update_quantity():
    name = input("Enter the name of the item to update quantity: ")
    if name in inventory:
        new_quantity = int(input("Enter the new quantity: "))
        inventory[name]['quantity'] += new_quantity
        print(f"Quantity of {name} updated to {inventory[name]['quantity']}.")
    else:
        print(f"{name} not found in the inventory.")

def view_inventory():
    print("Current Inventory:")
    for item, details in inventory.items():
        print(f"Item: {item}, Quantity: {details['quantity']}, Price: ${details['price']}")

def remove_item():
    name = input("Enter the name of the item to remove: ")
    if name in inventory:
        del inventory[name]
        print(f"{name} removed from the inventory.")
    else:
        print(f"{name} not found in the inventory.")

# Menu for the grocery store manager
while True:
    print("\nMenu:")
    print("1. Add new item")
    print("2. Update item quantity")
    print("3. View inventory")
    print("4. Remove item")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == '1':
        add_item()
    elif choice == '2':
        update_quantity()
    elif choice == '3':
        view_inventory()
    elif choice == '4':
        remove_item()
    elif choice == '5':
```

```
        print("Exiting inventory management system.")
        break
    else:
        print("Invalid choice. Please enter a number between 1 and 5.")
```

```
Menu:
1. Add new item
2. Update item quantity
3. View inventory
4. Remove item
5. Exit
Enter your choice (1-5): 1
Enter the name of the item: Biscuits
Enter the quantity of the item: 4
Enter the price of the item: 30
Biscuits added to the inventory.

Menu:
1. Add new item
2. Update item quantity
3. View inventory
4. Remove item
5. Exit
Enter your choice (1-5): 3
Current Inventory:
Item: Biscuits, Quantity: 4, Price: $30.0

Menu:
1. Add new item
2. Update item quantity
3. View inventory
4. Remove item
5. Exit
Enter your choice (1-5): 2
Enter the name of the item to update quantity: Bread
Bread not found in the inventory.

Menu:
1. Add new item
2. Update item quantity
3. View inventory
4. Remove item
5. Exit
Enter your choice (1-5): 5
Exiting inventory management system.
```