In [4]:

```
# Part 1:

# This function imports product data from a file and stores it in a dictionary.
def import_stock(filename):
    stock = {}  # Create an empty dictionary to store product data.

    try:
        with open(filename, 'r') as file:  # Open the specified file for reading.
            for line in file:  # Iterate over each line in the file.
                product, quantity = line.strip().split(',')  # Split each line into pr
                stock[product] = int(quantity)  # Add the product name and quantity to
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return stock  # Return an empty dictionary if an error occurs.

    return stock  # Return the populated dictionary containing product data.

# Call the import_stock function to read product data from the 'stock.txt' file.
stock = import_stock('stock.txt')  # The 'stock' variable now contains product data.

# Display the 'stock' dictionary to view the imported product data.
display(stock)
```

```
{'coke': 10, 'juice': 5, 'milk': 13}
```

In [5]:

```
# Part 2:

# Define a function to generate overall statistics for all products.
def gen_stats(stock):
    if not stock:
        print("No products in stock.")  # Check if the 'stock' dictionary is empty and
        return

    # Calculate the total, average, maximum, and minimum quantities for all products.
    total_quantity = sum(stock.values())  # Calculate the sum of all quantities.
    average_quantity = total_quantity / len(stock)  # Calculate the average quantity.
    max_quantity = max(stock.values())  # Find the maximum quantity.
    min_quantity = min(stock.values())  # Find the minimum quantity.

    # Display the overall statistics to the user.
    print("Overall Statistics:")
    print()
    print(f"Total amount for all products: {total_quantity}")
    print(f"Average amount for all products: {average_quantity:.2f}")  # Format the av
    print(f"Maximum amount for a product: {max_quantity}")
    print(f"Minimum amount for a product: {min_quantity}")

# Call the gen_stats function with the 'stock' dictionary to calculate and display the
gen_stats(stock)
```

```
Overall Statistics:

Total amount for all products: 28
Average amount for all products: 9.33
Maximum amount for a product: 13
Minimum amount for a product: 5
```

In [6]:

```
# Part 3:
```

```
# Define a function to check the stock quantity for a given product in a case-insensit
def check_stock(stock, product_name):
    product_name = product_name.lower()  # Convert the input to lowercase for case-ins
    if product_name in stock:  # Check if the converted product name exists in the sto
        return stock[product_name]  # Return the quantity for the matching product.
    else:
        return None  # Return None if the product is not found in stock.

# Ask the user to input a product name.
product_name = input("Enter a product name: ").lower()  # Convert the user's input to

# Call the check_stock function with the stock dictionary and the user's input.
quantity = check_stock(stock, product_name)

if quantity is not None:  # Check if the product was found in stock.
    # Print the product name with the original case and the corresponding quantity.
    print(f"The number in stock for {product_name.capitalize()} is: {quantity}")
else:
    print("We don't have this product.")  # Print a message if the product is not foun
```

```
Enter a product name: orange
We don't have this product.
```

In [7]:

```
# Part 4

# Define a function to update the stock for a given product.
def update_stock(stock, product_name, new_quantity):
    # Convert the product name to lowercase for case-insensitive comparison.
    product_name = product_name.lower()
    if product_name in stock:
        stock[product_name] = new_quantity
        return True  # Return True if an update was made.
    else:
        print(f"We don’t have the product: {product_name}")
        return False  # Return False if no update was made.

# Ask the user to input product names and quantities, separated by commas.
product_input = input("Enter product names and quantities (e.g., coke, 11, milk, 14):
product_pairs = product_input.split(',')  # Split the input by commas to get pairs of

# Process the pairs in a loop.
i = 0
updated = False  # To track if any updates were made.
while i < len(product_pairs):
    product_name = product_pairs[i].strip().lower()  # Convert the product name to low
    new_quantity = int(product_pairs[i + 1].strip())
    i += 2  # Advance by 2 to move to the next product pair.

    # Call the update_stock function and use a bitwise OR operation to combine update
    updated |= update_stock(stock, product_name, new_quantity)

# Print the updated dictionary if any updates were made.
if updated:
    print("The updated dictionary is:", stock)
```

```
Enter product names and quantities (e.g., coke, 11, milk, 14): coke, 11
The updated dictionary is: {'coke': 11, 'juice': 5, 'milk': 13}
```

In [ ]: