ASSESSMENT QUESTIONS

1. Create a program that prints the sum of 10 numbers.

2. Create a program that reads 3 numbers, and prints the output in this format

   The average of {num1}, {num2} and {num3} is {average}

   Where {num1}, {num2} and {num3} are the numbers entered by the user; and {average} is the average of the three numbers.

3. Create a program that allows the user to enter their name, age and phone number and output it in this format

   New User:

   Name: {name}

   Age: {age}

   Phone_number: {phone}

   Where {name} is the name entered by the user,

   {age} is the age entered by the user, and

   {phone} is the phone_number entered by the user.

4. Create a program that reads two strings from the user, concatenate them and return the output in this format

   The first string is {text1}, the second string is {text2} and the result of the concatenation is {concatenation_result}

   Where {text1} and {text2} are the strings entered by the user.

5. You will write a program that allows you to read details of a store inventory from the user and print them to the console.

   Read store name, address, and an item from the user and print it to the console in this format

   Store:

   Name: {name}

   Address: {address}

   Items:

   Item:

Name: {item_name}

SKU: {item_sku}

Quantity: {item_quantity}

Price: {item price}

Where {name} is the name entered by the user

{address} is the address of the store entered by the user

{item_name} is the name of the item (remember we are reading only one item)

{item_sku} is the SKU of the item

{item_quantity} is the quantity of that item that is left in the store

{item_price} is the price of the item.

6. (Compute the perimeter of a triangle) Write a program that reads three edges for a

   Triangle and computes the perimeter if the input is valid. Otherwise, display that

   The input is invalid. The input is valid if the sum of every pair of two edges is

   Greater than the remaining edge.

   Here is a sample run:

   Enter three edges: (user enters 1, 1, 1)

   The perimeter is 3

   Here is another:

   Enter three edges: (user enters 1, 3, 1)

   The input is invalid

7. (Special question) Extend your simple inventory management system from question 5 above solution to check certain

   Conditions to determine whether a user's input is valid. If it is not valid, notify the user and terminate the program.

   If it is, store the information in variables. You will also extend it to take up to 5 items depending on

   Whether the user wants to enter new items.

   The rules you will check are:

   - Name must not be empty

- Address must not be empty
- Number of items must be greater than or equal to 0
- Item name must not be empty
- Item SKU must not be empty
- Quantity must be greater than or equal to 0
- Item price must be greater than 0

Only the number of items the user wants to enter (up to a maximum of 5) must be printed

Store:

Name: {name}

Address: {address}

Items:

Item1:

Name: {item1_name}

SKU: {item1_sku}

Quantity: {item1_quantity}

Price: {item1_price}

Item2:

Name: {item2_name}

SKU: {item2_sku}

Quantity: {item2_quantity}

Price: {item2_price}

Item3:

Name: {item3_name}

SKU: {item3_sku}

Quantity: {item3_quantity}

Price: {item3_price}

Item4:

Name: {item4_name}

SKU: {item4_sku}

Quantity: {item4_quantity}

Price: {item4_price}

Item5:

Name: {item5_name}

SKU: {item5_sku}

Quantity: {item5_quantity}

Price: {item5_price}

Where {name} is the name entered by the user

{address} is the address of the store entered by the user

{itemi_name} is the name of the ith item ()

{itemi_sku} is the SKU of the ith item

{itemi_quantity} is the quantity of the ith item that is left in the store

{itemi_price} is the price of the ith item

Example result:

Store:

Name: Abdulrazaq Store

Address: No 8, Abeokuta Road

Items:

Item1:

Name: Golden Morn (1kg)

SKU: SAS-GM-1KG

Quantity: 23

Price: 1430.00

Item2:

Name: Golden Morn (500g)

SKU: SAS-GM-500G

Quantity: 11

Price: 800.00

Another example result:

Store:

Name: Olayemi Store

Address: No 8, Abeokuta Road

Items:

Item1:

Name: Golden Morn (1kg)

SKU: SAS-GM-1KG

Quantity: 23

Price: 1430.00

Item2:

Name: Golden Morn (500g)

SKU: SAS-GM-500G

Quantity: 11

Price: 800.00

Item3:

Name: OMO Detergent (50g)

SKU: SAS-OMO-50G

Quantity: 221

Price: 50.00

Item4:

Name: WOW Detergent (50g)

SKU: SAS-WOW-50G

Quantity: 301

Price: 50.00

Item5:

Name: Nasco Corn Flakes (500g)

SKU: SAS-NCF-500G

Quantity: 21

Price: 750.00

8. (Conversion from kilograms to pounds) Write a program

That displays the following two tables side by side (note that 1 kilogram is 2.205 Pounds):

Kilograms Pounds

1 2.21

3 6.62

...

197 433.38

199 437.80

9. (Conversion from miles to kilometers) Write a program

That displays the following two tables side by side (note that 1 mile is 1.609 kilometers):

Miles Kilometres

1 1.609

2 3.218

...

9 15.481

10 16.090

10. (Occurrence of max numbers) Write a program that reads integers, finds the

Largest of them, and counts its occurrences. Assume that the input ends with number 0. Suppose that you entered 3 5 2 5 5 5 0; the program finds that the

Largest number is 5 and the occurrence count for 5 is 4.

(Hint: Maintain two variables, max and count. The variable max stores the current maximum number, and

Count stores its occurrences. Initially, assign the first number to max and 1 to

Count. Compare each subsequent number with max. If the number is greater than

Max, assign it to max and reset count to 1. If the number is equal to max, increment count by 1.)

Sample run:

Enter a number (0: for end of input): (user enters 3)

Enter a number (0: for end of input): (user enters 5)

Enter a number (0: for end of input): (user enters 2)

Enter a number (0: for end of input): (user enters 5)

Enter a number (0: for end of input): (user enters 5)

Enter a number (0: for end of input): (user enters 5)

Enter a number (0: for end of input): (user enters 0)

The largest number is 5

The occurrence count of the largest number is 4