SECTION-A:

Q1:

Write a program that uses a print statement to say 'hello world' as shown in 'Desired Output'.

Desired Output

hello world

Q2:

Write a program that uses input to prompt a user for their name and then welcomes them. Note that input will pop up a dialog box. Enter Sarah in the pop-up box when you are prompted so your output will match the desired output.

Desired Output

Hello Pidugu

Q3:

Write a program to prompt the user for hours and rate per hour using input to compute gross pay.

Use 35 hours and a rate of 2.75 per hour to test the program (the pay should be 96.25). You should use input to read a string and float() to convert the string to a number. Do not worry about error checking or bad user data.

Desired Output

Pay: 96.25

Q4:

Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error. If the score is between 0.0 and 1.0, print a grade using the following table:

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

If the user enters a value out of range, print a suitable error message and exit. For the test, enter a score of 0.85.

Desired Output

B

Q5:

Write a program to prompt the user for hours and rate per hour using input to compute gross pay.

Pay the hourly rate for the hours up to 40 and 1.5 times the hourly rate for all hours worked above 40 hours.

Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75).

You should use input to read a string and float() to convert the string to a number.

Do not worry about error checking the user input - assume the user types numbers properly.

Desired Output

498.75

Q6:

Write a program to prompt the user for hours and rate per hour using input to compute gross pay.

Pay should be the normal rate for hours up to 40 and time-and-a-half for the hourly rate for all hours worked above 40 hours.

Put the logic to do the computation of pay in a function called computepay() and use the function to do the computation.

The function should return a value. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75).

You should use input to read a string and float() to convert the string to a number.

Do not worry about error checking the user input unless you want to - you can assume the user types numbers properly.

Do not name your variable sum or use the sum() function.

Desired Output

498.75

SECTION-B:

Question 1:

Write one for loop to print out each character of the string my_str on a separate line.

my_str = "MICHIGAN"

Question 2:

Write one for loop to print out each element of the list several_things. Then, write another for loop to print out the TYPE of each element of the list several_things. To complete this problem you should have written two different for loops, each of which iterates over the list several_things, but each of those 2 for loops should have a different result.

several_things = ["hello", 2, 4, 6.0, 7.5, 234352354, "the end", "", 99]

Question 3:

Write code that uses iteration to print out the length of each element of the list stored in str_list.

str_list = ["hello", "", "goodbye", "wonderful", "I love Python"]

Question 4:

Write code to count the number of characters in original_str using the accumulation pattern and assign the answer to a variable num_chars. Do NOT use the len function to solve the problem (if you use it while you are working on this problem, comment it out afterward!)

original_str = "The quick brown rhino jumped over the extremely lazy fox."

Question 5:

addition_str is a string with a list of numbers separated by the + sign. Write code that uses the accumulation pattern to take the sum of all of the numbers and assigns it to sum_val (an integer). (You should use the .split("+") function to split by "+" and int() to cast to an integer).

addition_str = "2+5+10+20"

Question 6:

week_temps_f is a string with a list of fahrenheit temperatures separated by the , sign. Write code that uses the accumulation pattern to compute the average (sum divided by number of items) and assigns it to avg_temp. Do not hard code your answer (i.e., make your code compute both the sum or the number of items in week_temps_f) (You should use the .split(",") function to split by "," and float() to cast to a float).

week_temps_f = "75.1,77.7,83.2,82.5,81.0,79.5,85.7"

Question 7:

Write code to create a list of numbers from 0 to 67 and assign that list to the variable nums. Do not hard code the list.

Question 8:
Write code to create a list of word lengths for the words in original_str using the accumulation pattern and assign the answer to a variable num_words_list. (You should use the len function).
original_str = "The quick brown rhino jumped over the extremely lazy fox"

Question 9:
Create an empty string and assign it to the variable lett. Then using range, write code such that when your code is run, lett has 7 b's ("bbbbbbb").

Question 10:
For each word in words, add 'd' to the end of the word if the word ends in "e" to make it past tense. Otherwise, add 'ed' to make it past tense. Save these past tense words to a list called past_tense.
words = ["adopt", "bake", "beam", "confide", "grill", "plant", "time", "wave", "wish"]

Question 11:
rainfall_mi is a string that contains the average number of inches of rainfall in Michigan for every month (in inches) with every month separated by a comma. Write code to compute the number of months that have more than 3 inches of rainfall. Store the result in the variable *num_rainy_months*. In other words, count the number of items with values > *3.0*.
rainfall_mi = "1.65, 1.46, 2.05, 3.03, 3.35, 3.46, 2.83, 3.23, 3.5, 2.52, 2.8, 1.85"

Question 12:
The variable *sentence* stores a string. Write code to determine how many words in *sentence* start and end with the same letter, including one-letter words. Store the result in the variable *same_letter_count*.
sentence = "students flock to the arb for a variety of outdoor activities such as jogging and picnicking"

Question 13:
Write code to count the number of strings in list *items* that have the character w in it. Assign that number to the variable acc_num.
HINT 1: Use the accumulation pattern!
HINT 2: the *in* operator checks whether a substring is present in a string.
items = ["whirring", "wow!", "calendar", "wry", "glass", "", "llama","tumultuous","owing"]

Question 14:
Write code that counts the number of words in sentence that contain either an "a" or an "e". Store the result in the variable num_a_or_e.
Note 1: be sure to not double-count words that contain both an a and an e.
HINT 1: Use the in operator.
HINT 2: You can either use or or elif.
sentence = "python is a high level general purpose programming language that can be applied to many different classes of problems."

Question 15:
Write code that will count the number of vowels in the sentence s and assign the result to the variable num_vowels. For this problem, vowels are only a, e, i, o, and u. Hint: use the in operator with vowels.
s = "singing in the rain and playing in the rain are two entirely different situations but both can be fun"
vowels = ['a','e','i','o','u']

Question 16:
Write a program that repeatedly prompts a user for integer numbers until the user enters 'done'.
Once 'done' is entered, print out the largest and smallest of the numbers.
If the user enters anything other than a valid number catch it with a try/except and put out an appropriate message
and ignore the number. Enter 7, 2, bob, 10, and 4 and match the output below.
Desired Output
Invalid input
Maximum is 10
Minimum is 2