**Answer Key for Exercises in Python Resource Book**

**A5. The output and reasoning for each Python command:**

1. **print(‘One step at a time’)**
   * **Output**: One step at a time
   * **Reason**: The print function outputs the string 'One step at a time'.
2. **print(“What’s the next step?”)**
   * **Output**: What’s the next step?
   * **Reason**: The print function outputs the string "What’s the next step?".
3. **print(“What’s the next step?’)**
   * **Output**: **Syntax Error**
   * **Reason**: There is a mismatch in the quotation marks. Python requires consistent use of either single or double quotes to enclose a string.
4. **A = 3.4**
   * **Reason**: The variable A is assigned the float value 3.4.
5. **A = 2 + 3.0**
   * **Reason**: The variable A is reassigned the result of 2 + 3.0, which is 5.0.
6. **print(A)**
   * **Output**: 5.0
   * **Reason**: The print function outputs the current value of A, which is 5.0.
7. **9/4**
   * **Output**: 2.25
   * **Reason**: Division of integers 9/4 in Python 3 results in a float.
8. **9/4.0**
   * **Output**: 2.25
   * **Reason**: Division of an integer by a float results in a float.
9. **9.0/2**
   * **Output**: 4.5
   * **Reason**: Division of a float by an integer results in a float.
10. **B = A**
    * **Reason**: The variable B is assigned the current value of A, which is 5.0.
11. **B = 3**
    * **Reason**: The variable B is reassigned the integer value 3.
12. **B == A**
    * **Output**: False
    * **Reason**: B (which is now 3) is not equal to A (which is 5.0), so the comparison returns False.
13. **17 % 7**
    * **Output**: 3
    * **Reason**: The modulo operator % returns the remainder of the division of 17 by 7, which is 3.
14. **17.0 % 7**
    * **Output**: 3.0
    * **Reason**: The modulo operation between a float and an integer returns a float remainder, which is 3.0.
15. **‘Ha’ + ‘Ha’ + ‘Ha’**
    * **Output**: HaHaHa
    * **Reason**: String concatenation adds the strings together.
16. **‘Ha’\*3**
    * **Output**: HaHaHa
    * **Reason**: String multiplication repeats the string 'Ha' three times.

**A.7 Practice Time - Sequential Construct**

1. Write a program to accept 2 numbers of float data type and find the difference A-B.

# Accepting two float numbers

A = float(input("Enter the first number (A): "))

B = float(input("Enter the second number (B): "))

# Calculating and printing the difference

difference = A - B

print("The difference A - B is:", difference)

2. Write a program to accept 3 numbers and print the product of the three numbers.

# Accepting three numbers

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

num3 = float(input("Enter the third number: "))

# Calculating and printing the product

product = num1 \* num2 \* num3

print("The product of the three numbers is:", product)

3. Write a program to accept the length and breadth of a rectangle and find the area and perimeter.

# Accepting length and breadth of the rectangle

length = float(input("Enter the length of the rectangle: "))

breadth = float(input("Enter the breadth of the rectangle: "))

# Calculating and printing the area and perimeter

area = length \* breadth

perimeter = 2 \* (length + breadth)

print("Area of the rectangle is:", area)

print("Perimeter of the rectangle is:", perimeter)

4. Write a program to accept radius measurement of a circle and print its area and circumference.

# Accepting radius of the circle

radius = float(input("Enter the radius of the circle: "))

# Calculating and printing the area and circumference

area = 3.14159 \* radius \* radius

circumference = 2 \* 3.14159 \* radius

print("Area of the circle is:", area)

print("Circumference of the circle is:", circumference)

**B.3 Practice Exercise - Simple IF statements**

.**1. Write a program that accepts a number and checks if it is divisible by 3**

# Accepting a number

num = int(input("Enter a number: "))

# Checking if the number is divisible by 3

if num % 3 == 0:

print("The number “, num, “is divisible by 3.")

else:

print("The number “, num, “is not divisible by 3.")

**2. Write a program to check if a given number is odd or even.**

# Accepting a number

num = int(input("Enter a number: "))

# Checking if the number is odd or even

if num % 2 == 0:

print("The number “, num, “is even.")

else:

print("The number “, num, “is odd.")

**3. Write a program that accepts 2 numbers and prints the smaller number.**

# Accepting two numbers

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

# Finding and printing the smaller number

if num1 < num2:

print("The smaller number is: “,num1)

else:

print(f"The smaller number is: “, num2)

**4. In the previous program, what happens if both the numbers are equal? Justify your answer.**

# Accepting two numbers

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

# Checking and printing the smaller number or if they are equal

if num1 < num2:

print("The smaller number is: “,num1)

elif num2 < num1:

print("The smaller number is: “, num2)

else:

print("Both numbers are equal: “, num1)

### **Explanation and Behavior:**

1. If num1 is smaller than num2:
   * The first if condition (num1 < num2) evaluates to True, and the program prints num1 as the smaller number.
2. If num2 is smaller than num1:
   * The elif condition (num2 < num1) evaluates to True, and the program prints num2 as the smaller number.
3. If both numbers are equal:
   * Neither if nor elif conditions are satisfied. The else block executes, printing "Both numbers are equal.".

### **Justification:**

The else block is designed to handle the scenario where both numbers are equal. Since no number is smaller in such cases, the output appropriately states that the two numbers are equal.

**B4. Practice Exercise:**

**1. Write a program that accepts a number from 1 to 7 and displays the corresponding day of the week. If any other number is entered, program should display the message, “Invalid Number “.**

# Accepting a number from 1 to 7

day\_number = int(input("Enter a number from 1 to 7: "))

# Displaying the corresponding day of the week

if day\_number == 1:

print("Monday")

elif day\_number == 2:

print("Tuesday")

elif day\_number == 3:

print("Wednesday")

elif day\_number == 4:

print("Thursday")

elif day\_number == 5:

print("Friday")

elif day\_number == 6:

print("Saturday")

elif day\_number == 7:

print("Sunday")

else:

print("Invalid Number")

**C1. Practice Exercise:**

| **Command** | **Output** | **Reason** |
| --- | --- | --- |
| len(5) | TypeError | len() expects a collection (e.g., list, string, tuple) but 5 is an integer and not iterable. |
| len([1, 2, 3, 4]) | 4 | len() returns the number of elements in the list [1, 2, 3, 4], which is 4. |
| len('apple') | 5 | len() returns the number of characters in the string 'apple', which is 5. |
| max(-9, 5, 0.5) | 5 | max() returns the largest value among -9, 5, and 0.5, which is 5. |
| max([1, 2, 3], 5.6) | TypeError | The max() function cannot compare a list ([1, 2, 3]) and a float (5.6) as they are different data types. |
| min(-22, -2, 22) | -22 | min() returns the smallest value among -22, -2, and 22, which is -22. |
| list(range(2, 21, 3)) | [2, 5, 8, 11, 14, 17, 20] | range(2, 21, 3) generates numbers starting from 2 to less than 21, incrementing by 3. |
| list(range(4)) | [0, 1, 2, 3] | range(4) generates numbers starting from 0 to less than 4. |
| list(range(10, 4, -2)) | [10, 8, 6] | range(10, 4, -2) generates numbers starting from 10 to greater than 4, decrementing by -2. |
| list(range()) | [] | An empty range() generates an empty list because no start, stop, or step values are provided. |
| round(33.336, 2) | 33.34 | round() rounds 33.336 to 2 decimal places, resulting in 33.34 due to rounding up at the third decimal place (6). |
| round(3.141592, 2) | 3.14 | round() rounds 3.141592 to 2 decimal places, resulting in 3.14 as the third decimal place (1) does not require rounding up. |

**D.3 Practice Exercise - FOR loop**

**1. Write a program to display the first 7 multiples of 35.**

# Displaying the first 7 multiples of 35

for i in range(1, 8):

print(35 \* i)

**2. Find the sum of the first 10 natural numbers.**

# Finding the sum of the first 10 natural numbers

sum\_of\_numbers = 0

for i in range(1, 11):

sum\_of\_numbers += i

print("The sum of the first 10 natural numbers is:", sum\_of\_numbers)

**3. Accept the marks of 5 subjects from the user as float values and find the average of these marks.**

# Accepting marks of 5 subjects and calculating the average

total\_marks = 0

for i in range(1, 6):

marks = float(input("Enter the marks for subject”,i, ”,: "))

total\_marks += marks

average\_marks = total\_marks / 5

print("The average marks are:", average\_marks)

**4. Print the following pattern:**

**1 1 1 1 1**

**2 2 2 2 2**

**3 3 3 3 3**

# Printing the pattern

for i in range(1, 4):

for j in range(5):

print(i, end=" ")

print()

**5. Given that l = [1,3,5,7,9], print 2, 4, 6, 8, 10 using a FOR loop.**

# Given list

l = [1, 3, 5, 7, 9]

# Printing the next even numbers in the sequence

for i in range(1, 6):

print(l[i-1] + 1, end=" ")

**D.5 Practice Exercises for While Loop**

**1. Write a program that simulates a card game. The user inputs a card number from 1 to 13. If the input number is 13, the user does not get another card. If the input number is negative, then the player does not get another card.**

# Simulating a card game

card\_number = 0

while True:

card\_number = int(input("Enter a card number (1 to 13): "))

if card\_number == 13:

print("You got the highest card. No more cards for you!")

break

elif card\_number < 0:

print("Invalid input. Game over!")

break

else:

print("You got card number is,” , card\_number, “.Draw another card!”)

**2. Write a program that keeps accepting the age of employees in a loop. If the age entered is negative or above 57, then the loop should stop executing. The program should also print the number of employees who are in the age group 25 to 35.**

# Initializing variables

employee\_count = 0

age\_count = 0

# Loop to accept employee ages

while True:

age = int(input("Enter the employee's age: "))

# Stop the loop if age is negative or above 57

if age < 0 or age > 57:

break

# Count employees in age group 25-35

if age>= 25 and age<=35:

age\_count = age\_count + 1

employee\_count = employee\_count+1

# Printing the results

print("Number of employees in the age group 25-35: ”, age\_count)

**3. Write a program that calculates the factorial of a number using a while loop. (Factorial of 4 is**

**1\*2\*3\*4, Factorial of 6 is 1 \*2\*3\*4\*5\*6).**

# Accepting a number

number = int(input("Enter a number to calculate its factorial: "))

# Initializing variables

factorial = 1

i = 1

# Calculating factorial using a while loop

while i <= number:

factorial = factorial \* ii

i = i + 1

# Printing the result

print("The factorial of”, number, “is: ”, factorial)

**Unit End Exercises:**

1. **Correct the errors in the given programs:**
2. 

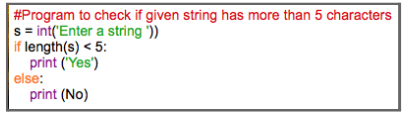
Answer:

* 1. The program is meant to print the value of pi up to **5 decimal places**, but round(p, 2) rounds the value to **2 decimal places** instead of 5.
  2. Brackets in the print function are missing.
  3. The corrected code is:

# Program to print pi up to 5 decimal places

p = 22 / 7

print(round(p, 5))

1. ****

Answer:

* 1. **Incorrect Conversion of Input (int()):**
     1. The code s = int('Enter a string') attempts to convert the string 'Enter a string' into an integer, which is not correct. Instead, you should be using input() to accept a string from the user.
  2. **Incorrect Function Name (length()):**
     1. The function length(s) is not valid in Python. To find the length of a string, you should use the built-in function len(s).
  3. **Syntax Error in Print Statement (No without quotes):**
     1. In the print(No), "No" should be enclosed in quotes to make it a valid string.

### **Corrected Code:**

# Program to check if given string has more than 5 characters

s = input("Enter a string: ") # Using input() to get a string

if len(s) > 5: # Checking if the length of the string is more than 5

print("yes")

else:

print("No")

1. 

Answer:

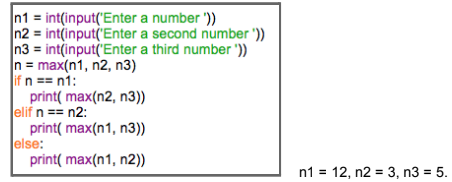
**Accessing List Elements Incorrectly:**

* In the loop for i in L:, i takes on the values of the list elements, not the indices. However, the code attempts to access L[i], which will cause an error since i is an element, not an index.
* If i is 1, L[i] would refer to L[1], but you need to directly multiply the element i itself.

**Using Index Instead of Element:**

* You should directly multiply the value i by 3 rather than using L[i].

**B. Predict the output of the given programs and state the purpose of the program.**

1. 

Answer: The program prints the second largest item in the list: “5”

1. 

Answer:

The output is:

3

6

9

12

15

18

21

24

27

30

1. 

Answer: Since the while loop condition is False, the body of the loop will **never** execute. In Python, a while loop runs as long as the condition evaluates to True. In this case, because the condition is False, the loop will be skipped entirely.

### **Output:**

There will be **no output** at all, as the loop does not run.

**C. Describe these functions:**

* 1. **print()**: The print() function outputs specified text, variables, or expressions to the console. It automatically moves to the next line after printing.
  2. **input()**: The input() function prompts the user for input, returning it as a string. It pauses program execution until the user provides a response.
  3. **int()**: The int() function converts a specified value into an integer. It can handle string representations of numbers or floating-point values for conversion.
  4. **max()**: The max() function returns the largest item from an iterable or the largest of two or more arguments provided to it.
  5. **round()**: The round() function rounds a floating-point number to a specified number of decimal places, returning the result as a rounded number.

**D. Answer the following.**

1. **When do we need to use IF-ELIF-ELSE block?**

Ans: Use the if-elif-else block when you need to check multiple conditions sequentially. It allows you to execute different blocks of code based on varying conditions.

1. **Explain range() with examples.**
2. Ans: The range() function generates a sequence of numbers within a specified range. It can take one, two, or three arguments: range(start, stop, step). Example: range(1, 10, 2) generates 1, 3, 5, 7, 9.
3. **What is the difference between FOR and WHILE loop?**

Ans: A for loop iterates over a sequence or iterable, like a list or range. A while loop continues until a specified condition becomes False, providing more flexibility for dynamic conditions.

1. **Write the syntax for while loop:**

Ans: The syntax for a while loop is:

while <condition>:

# code to execute as long as condition is True

Example: while x < 5: print(x); x += 1

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***