

Q 1. WAP in Python to Take Three Numbers and Print Their Sum

python

Copy code

```
# Taking three numbers as input

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))


# Calculating the sum

total = num1 + num2 + num3


# Printing the sum

print(f"The sum of {num1}, {num2}, and {num3} is {total}.")
```

Q 2. WAP in Python to Calculate the Area of a Rectangle

```
# Taking length and breadth as input

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


# Calculating area

area = length * breadth


# Printing the area

print(f"The area of the rectangle is {area} square units.")
```

Q 3. WAP in Python to Print Invoice with GST, CGST, and SGST

```
# Taking input for selling price and GST rate

selling_price = float(input("Enter the selling price: "))
gst_rate = float(input("Enter the GST rate (as %): "))


# Calculating CGST and SGST

gst_amount = (selling_price * gst_rate) / 100
```

```
cgst = sgst = gst_amount / 2

# Printing the invoice

print("\n--- Invoice ---")

print(f"Selling Price: ₹{selling_price:.2f}")

print(f"GST Rate: {gst_rate}%")

print(f"CGST: ₹{cgst:.2f}")

print(f"SGST: ₹{sgst:.2f}")

print(f"Total Price: ₹{selling_price + gst_amount:.2f}")
```

Q 4. WAP in Python to Display a Joke with a Conditional Punch Line

```
# Display the joke setup

print("Why don't skeletons fight each other?")

input("Press Enter to reveal the punchline...")

# Display the punchline

print("Because they don't have the guts!")
```

Q 5. WAP in Python to Convert Height from Centimeters to Feet and Inches

```
# Taking height in centimeters as input

height_cm = float(input("Enter your height in centimeters: "))

# Conversion factors

cm_to_inches = 0.393701

cm_to_feet = 0.0328084

# Calculating height in feet and inches

height_inches = height_cm * cm_to_inches

feet = int(height_inches // 12)

inches = round(height_inches % 12, 2)

# Printing the converted height

print(f"Your height is approximately {feet} feet and {inches} inches.")
```

Output of the Provided Code

Code:

```
a = 5
b = -3
c = 25
d = -10

print(a + b + c > a + c - b * d)
print(str(a + b + c > a + c - b * d) == 'true')
print(len(str(a + b + c > a + c - b * d)) == len(str(bool(1))))
```

Step-by-Step Execution:

1. **Expression 1:** $a + b + c > a + c - b * d$
 - Left-hand side: $a + b + c = 5 + (-3) + 25 = 27$
 - Right-hand side: $a + c - b * d = 5 + 25 - (-3) * (-10) = 5 + 25 - 30 = 0$
 - $27 > 0$ is True.
2. **Expression 2:** $\text{str}(a + b + c > a + c - b * d) == \text{'true'}$
 - $\text{str}(\text{True})$ is "True" (case-sensitive).
 - $\text{"True"} == \text{'true'}$ is False (because of case mismatch).
3. **Expression 3:** $\text{len}(\text{str}(a + b + c > a + c - b * d)) == \text{len}(\text{str}(\text{bool}(1)))$
 - $\text{len}(\text{str}(\text{True})) = \text{len}(\text{"True"}) = 4$.
 - $\text{bool}(1)$ is True; $\text{len}(\text{str}(\text{bool}(1))) = \text{len}(\text{"True"}) = 4$.
 - $4 == 4$ is True.

Output:

True

False

True

Predict the Output of the Given Code:

Code:

```
a = 3 + 5 / 8
```

```
b = int(3 + 5 / 8)
c = 3 + float(5 / 8)
d = 3 + float(5) / 8
e = 3 + 5.0 / 8
f = int(3 + 5 / 8.0)
print(a, b, c, d, e, f)
```

Explanation:

1. **a = 3 + 5 / 8:**

- $5 / 8 = 0.625$ (floating point division).
- $a = 3 + 0.625 = 3.625$.

2. **b = int(3 + 5 / 8):**

- Same as a for $3 + 5 / 8 = 3.625$.
- `int(3.625)` converts it to 3 (by truncating the decimal part).

3. **c = 3 + float(5 / 8):**

- $5 / 8 = 0.625$.
- `float(0.625)` is 0.625 (no change).
- $c = 3 + 0.625 = 3.625$.

4. **d = 3 + float(5) / 8:**

- `float(5)` is 5.0.
- $5.0 / 8 = 0.625$.
- $d = 3 + 0.625 = 3.625$.

5. **e = 3 + 5.0 / 8:**

- $5.0 / 8 = 0.625$.
- $e = 3 + 0.625 = 3.625$.

6. **f = int(3 + 5 / 8.0):**

- $5 / 8.0 = 0.625$ (floating point division).
- $3 + 0.625 = 3.625$.
- `int(3.625)` converts it to 3.

Output:

3.625 3 3.625 3.625 3.625 3

Guess the Output for Boolean Conversions:

Code Snippets:

- **a. bool(0):**
 - bool(0) evaluates to False (zero is considered False in Python).
- **b. bool(1):**
 - bool(1) evaluates to True (non-zero is considered True in Python).
- **c. bool('0'):**
 - bool('0') evaluates to True (non-empty strings are considered True, even if they contain '0').
- **d. bool('1'):**
 - bool('1') evaluates to True (non-empty strings are considered True, even if they contain '1').
- **e. bool(''):**
 - bool('') evaluates to False (empty string is considered False in Python).
- **f. bool(0.0):**
 - bool(0.0) evaluates to False (zero is considered False, even for floating point).
- **g. bool('0.0'):**
 - bool('0.0') evaluates to True (non-empty string is considered True, even if it represents '0.0').
- **h. bool(0j):**
 - bool(0j) evaluates to False (zero complex number is considered False).
- **i. bool('0j'):**
 - bool('0j') evaluates to True (non-empty string is considered True, even if it represents '0j').

Summary of Outputs:

- **a. bool(0) → False**
- **b. bool(1) → True**
- **c. bool('0') → True**

- **d. `bool('1') → True`**
- **e. `bool('') → False`**
- **f. `bool(0.0) → False`**
- **g. `bool('0.0') → True`**
- **h. `bool(0j) → False`**
- **i. `bool('0j') → True`**

Q 6. WAP in Python to Obtain Principal Amount, Rate of Interest, and Time, Then Compute Simple Interest

Input principal amount, rate of interest, and time from the user

P = float(input("Enter the principal amount: "))

R = float(input("Enter the rate of interest (in %): "))

T = float(input("Enter the time period (in years): "))

Calculate simple interest

SI = (P * R * T) / 100

Print the result

print(f"The Simple Interest is: {SI}")

Q 7. WAP in Python to Calculate EMI (Equated Monthly Installment)

Formula for EMI:

$$E = \frac{P \times R \times (1+R)^n}{(1+R)^n - 1}$$

Where:

- P = Principal loan amount
- R = Monthly rate of interest (annual rate divided by 12)
- n = Loan tenure in months

Input principal, annual interest rate, and loan tenure in years

P = float(input("Enter the principal loan amount: "))

annual_rate = float(input("Enter the annual interest rate (in %): "))

```

n_years = int(input("Enter the loan tenure in years: "))

# Convert annual rate to monthly rate

R = annual_rate / 100 / 12

# Convert years to months

n = n_years * 12

# Calculate EMI


$$E = (P * R * (1 + R)^n) / ((1 + R)^n - 1)$$


# Print the result

print(f"The EMI is: {E:.2f}")

```

Q 8. WAP in Python that Reads Two Numbers and an Arithmetic Operator, Then Displays the Computed Result

```

# Input two numbers and an arithmetic operator from the user

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
operator = input("Enter an arithmetic operator (+, -, *, /): ")

# Compute and display the result based on the operator

if operator == "+":
    print(f"The result is: {num1 + num2}")
elif operator == "-":
    print(f"The result is: {num1 - num2}")
elif operator == "*":
    print(f"The result is: {num1 * num2}")
elif operator == "/":
    if num2 != 0:
        print(f"The result is: {num1 / num2}")
    else:
        print("Error! Division by zero.")

```

else:

print("Invalid operator.")

Q 9. WAP in Python to Calculate the Sum of Natural Numbers from 1 to 7, Printing Sum Progressively

```
sum_so_far = 0
```

```
# Loop through natural numbers from 1 to 7
```

```
for i in range(1, 8):
```

```
    sum_so_far += i
```

```
    print(f"Sum so far after adding {i}: {sum_so_far}")
```

Q 10. WAP in Python to Calculate the Average Income of a Shopkeeper for a Week

```
# Initialize variables to calculate total income
```

```
total_income = 0
```

```
# Input sales data for each day of the week
```

```
for i in range(1, 8):
```

```
    items_sold = int(input(f"Enter the number of items sold on day {i}: "))
```

```
    cash_received = float(input(f"Enter the total cash received on day {i}: "))
```

```
    total_income += cash_received
```

```
# Calculate average income for the week
```

```
average_income = total_income / 7
```

```
# Print the result
```

```
print(f"The average income of the shopkeeper for the week is: {average_income:.2f}")
```

Q 11. WAP in Python to Illustrate the Difference Between break and continue Statements

```
# Example of break and continue in a loop
```

```
print("Using break statement:")
```

```
for i in range(1, 11):
```

```
    if i == 6:
```



```
        break # Exit the loop when i is 6
    print(i)
print("\nUsing continue statement:")
for i in range(1, 11):
    if i == 6:
        continue # Skip printing 6
    print(i)
```

Q 12 . WAP in Python to Input Numbers Repeatedly and Finally Print Their Sum (Ends When 'no' or a Negative Number is Entered)

```
total_sum = 0
while True:
    user_input = input("Enter a number (or type 'no' to stop): ")
    if user_input.lower() == 'no':
        break # Exit the loop if user types 'no'
    number = float(user_input)
    if number < 0:
        break # Exit the loop if a negative number is entered
    total_sum += number
print(f"The total sum of the numbers entered is: {total_sum}")
```

Q 13 . WAP in Python to Display a String Given by User Like This:

```
# Input string from the user
string = input("Enter a string: ")
# Display the string in the required format
for i in range(len(string)):
    for j in range(i, len(string)):
        print(f"{string[i]} {string[j]}")
```

Q 14. WAP in Python that Reads a Line and a Substring, Then Displays the Number of Occurrences of the Substring in the Line

```
# Input line and substring from the user
line = input("Enter a line of text: ")
substring = input("Enter the substring: ")

# Count occurrences of the substring in the line
count = line.count(substring)

# Print the result
print(f"The substring '{substring}' appears {count} times in the line.")
```

Q 15. WAP in Python that Capitalizes Every Other Character in a String

```
# Input string from the user
string = input("Enter a string: ")

# Create a new string where every other character is capitalized
result = ""

for i in range(len(string)):
    if i % 2 == 0:
        result += string[i].lower() # Lowercase the character at even index
    else:
        result += string[i].upper() # Uppercase the character at odd index

# Print the result
print(f"Transformed string: {result}")
```

Q 16. WAP in Python that Checks Whether a String is a Palindrome

```
# Input string from the user
string = input("Enter a string: ")

# Check if the string is a palindrome
if string == string[::-1]:
```

```
print(f"The string '{string}' is a palindrome.")
else:
    print(f"The string '{string}' is not a palindrome.")
```

Q 17. Identify the Problem with the Following Code Fragment

Code:

```
S1 = 'must'
S2 = 'try'
N1 = 10
N2 = 3

print(S1 + S2) # Concatenates strings 'must' and 'try'
print(S2 * N2) # Repeats 'try' 3 times: 'trytrytry'
print(S1 + N1) # Error: Cannot concatenate string 'must' with integer 10
print(S2 * S1) # Error: Cannot repeat string 'try' by an integer value of 'must' (a string, not a number)
```

Explanation:

- S1 + S2 is valid as it concatenates two strings.
- S2 * N2 is valid, it repeats the string S2 3 times.
- S1 + N1 will result in an error because Python does not support concatenating a string and an integer. You need to convert N1 to a string: str(N1).
- S2 * S1 will also result in an error because S1 is a string, and Python expects an integer to repeat a string, not another string. You need to use an integer to repeat a string.

Fixing the code:

```
S1 = 'must'
S2 = 'try'
N1 = 10
N2 = 3

print(S1 + S2) # Concatenates strings 'must' and 'try'
print(S2 * N2) # Repeats 'try' 3 times: 'trytrytry'
print(S1 + str(N1)) # Converts N1 to string and concatenates
```

```
print(S2 * N2) # Corrected: repeating 'try' 3 times
```

Q 18. Output Produced by the Code with Different Inputs

Code:

```
String = input("Enter a string: ")
Count = 3
while True:
    if String[0] == 'a':
        String = String[2:] # Remove the first 2 characters
    elif String[-1] == 'b':
        String = String[:2] # Keep only the first two characters
    else:
        Count += 1
        break
print(String)
print(Count)
```

1. Input: aabbcc

- The first condition `String[0] == 'a'` is True, so the string becomes 'bbcc'.
- Next, `String[0] == 'b'`, so the loop continues.
- Now, `String[0] == 'b'` is False, and the condition else is met.
- Count is increased to 4, and the loop breaks.

Output:

bcc

4

2. Input: aaccbb

- The first condition `String[0] == 'a'` is True, so the string becomes 'ccbb'.
- Next, `String[0] == 'b'`, so the loop continues.
- Now, `String[0] == 'c'`, so Count becomes 4, and the loop breaks.

Output:

ccbb

4

3. Input: abcc

- The first condition `String[0] == 'a'` is True, so the string becomes 'cc'.
- Now, the string's first character is 'c', which does not meet any conditions, so Count increases to 4, and the loop breaks.

Output:

cc

4

5. Code Explanation with Conditions

Code:

```
Inp = input("Please enter a string: ")
```

```
while len(Inp) <= 4:
```

```
    if Inp[-1] == 'z':
```

```
        Inp = Inp[0:3] + 'c'
```

```
    elif 'a' in Inp:
```

```
        Inp = Inp[0] + 'bb'
```

```
    elif not int(Inp[0]):
```

```
        Inp = '1' + Inp[1:] + 'z'
```

```
    else:
```

```
        Inp = Inp + '*'
```

```
print(Inp)
```

Explanation:

- The while loop runs until the length of Inp is greater than 4.
- Inside the loop, based on the condition, different modifications are made to the string Inp:
 - If the last character is 'z', the string gets sliced and modified.
 - If the string contains 'a', it replaces all but the first character with 'bb'.
 - If the first character is not an integer, it replaces the first character and adds 'z' at the end.

- Otherwise, it adds '*' to the end of the string.

Examples:

1. Input: abc

- Length is 3, so loop runs.
- The string contains 'a', so the string becomes 'abb'.
- Length becomes 3 again, loop continues with the modified string.
- The condition elif 'a' in Inp will replace 'abb' to 'ab', and so on.

2. Input: 123

- Length is 3, so loop runs.
- Since the first character is not an integer, it modifies the string as described.

Q 19. Output Produced for the Input Strings

Code Snippet:

```
Inp = input("Please enter a string: ")
```

```
while len(Inp) <= 4:
```

```
    if Inp[-1] == 'z':
```

```
        Inp = Inp[0:3] + 'c'
```

```
    elif 'a' in Inp:
```

```
        Inp = Inp[0] + 'bb'
```

```
    elif not int(Inp[0]):
```

```
        Inp = '1' + Inp[1:] + 'z'
```

```
    else:
```

```
        Inp = Inp + '*'
```

```
print(Inp)
```

1. Input: 1bzz

- Length is 4, so the loop won't run.
- Output: 1bzz

2. Input: 1a

- Length is 2, so the loop will run.

- Inp = '1a' contains 'a', so it will modify Inp = '1bb'.
- Length is still 3, so the loop continues, and the condition 'a' in Inp will modify it to '1bb'.
- The process repeats until the length is greater than 4, resulting in Inp becoming 1bb*.
- Output: 1bb*

3. Input: abc

- Length is 3, so the loop will run.
- Inp = 'abc' contains 'a', so it becomes 'abb'.
- Length remains 3, so the loop continues, and the condition 'a' in Inp modifies Inp to 'ab'.
- The length eventually increases to greater than 4, so the loop ends.
- Output: ab*

4. Input: 0xy

- Length is 3, so the loop will run.
- Inp = '0xy' contains no 'a', and the last character is not 'z', so it goes to the else part and appends * to the string, making it '0xy*'.
- Length becomes 4, so the loop ends.
- Output: 0xy*

5. Input: xyz

- Length is 3, so the loop will run.
- Inp = 'xyz' contains no 'a' and doesn't end in 'z', so it appends * to the string, making it 'xyz*'.
- Length becomes 4, so the loop ends.
- Output: xyz*

Q 20. WAP to Find the Minimum Element from a List Along with Its Index

Input list from the user

```
lst = list(map(int, input("Enter a list of numbers: ").split()))
```

Find the minimum element and its index

```
min_value = min(lst)
```

```
min_index = lst.index(min_value)
```

Print the result

```
print(f"The minimum element is {min_value} at index {min_index}.")
```

Q 21. WAP to Calculate the Mean of a Given List Input by User

```
# Input list from the user
lst = list(map(int, input("Enter a list of numbers: ").split()))
# Calculate the mean
mean = sum(lst) / len(lst)
# Print the result
print(f"The mean of the list is: {mean}")
```

Q 22. WAP to Search for an Element in a Given List of Numbers

```
# Input list and element to search
lst = list(map(int, input("Enter a list of numbers: ").split()))
element = int(input("Enter the element to search for: "))
# Search for the element
if element in lst:
    print(f"Element {element} found at index {lst.index(element)}.")
else:
    print(f"Element {element} not found.")
```

Q 23. Examine the Given Code

Code:

```
Numlist = eval(input("Enter list: "))
Pos = 0
Odds = evens = 0
Length = len(Numlist)
while Pos < Length:
    if Numlist[Pos] % 2 == 0:
        evens = evens + 1
```



```
else:
    Odds = Odds + 1
    Pos = Pos + 1
print(evens)
print(Odds)
```

What is this program calculating?

This program counts the number of even and odd numbers in a given list Numlist.

- evens counts the number of even numbers.
- Odds counts the number of odd numbers.

What will be the output for input = [1, 5, 2, 3, 6, 6, 9]?

- Even numbers: 2, 6, 6 → 3 even numbers
- Odd numbers: 1, 5, 3, 9 → 4 odd numbers

Output:

3

4

Q 24. Consider the Following Code Fragments

Code a:

```
D1 = {'Bhavna': 1, 'Richard': 2, 'Firoza': 10, 'Dipa': 20}
temp = 0
for value in D1.values():
    temp = temp + value
print(temp)
```

- This code calculates the sum of all values in the dictionary D1.
- **Output:** 33 (1 + 2 + 10 + 20)

Code b:

```
D1 = {'Bhavna': 1, 'Richard': 2, 'Firoza': 10, 'Dipa': 20}
temp = " "
for key in D1:
```

```
if temp < key:
```

```
    temp = key
```

```
print(temp)
```

- This code finds the lexicographically largest key in the dictionary.
- **Output:** 'Richard' (since R comes lexicographically after the other letters)

Code c:

```
D1 = {'Bhavna': 1, 'Richard': 2, 'Firoza': 10, 'Dipa': 20}
```

```
k = 'Bhavna'
```

```
v = -1
```

```
if k in D1:
```

```
    D1[k] = v
```

```
print(D1)
```

- This code changes the value of the key 'Bhavna' to -1.
- **Output:** {'Bhavna': -1, 'Richard': 2, 'Firoza': 10, 'Dipa': 20}

Q 25. Write a Program in Python that Takes a Value and Checks Whether the Given Value is Part of a Dictionary

```
# Input dictionary and value to search
```

```
D1 = {'Bhavna': 1, 'Richard': 2, 'Firoza': 10, 'Dipa': 20}
```

```
value = int(input("Enter a value to search in the dictionary: "))
```

```
# Check if the value exists in the dictionary
```

```
if value in D1.values():
```

```
    # Find the corresponding key
```

```
    key = list(D1.keys())[list(D1.values()).index(value)]
```

```
    print(f"The value {value} corresponds to the key '{key}'.")
```

```
else:
```

```
    print("Error: The value is not found in the dictionary.")
```