Name: Kamlesh Kumar Sah
Batch: 36'B'

for-loop

May 27, 2024

0.1 Solve these questions using for loop

0.1.1 1. print “softwarica” 10 times

[32]:
```
for i in range(10):
    print("softwarica")
```

```
softwarica
softwarica
softwarica
softwarica
softwarica
softwarica
softwarica
softwarica
softwarica
softwarica
```

0.2 2. Sum of a list

[33]:
```
lst = [1, 2, 3, 4, 5]
total = 0
for num in lst:
    total += num
print(total)
```

```
15
```

1 3. print each character using indexing

[34]:
```
s = "softwarica"
for i in range(len(s)):
    print(s[i])
```

```
s
o
f
t
```

```
w
a
r
i
c
a
```

1.1 4. write a program to display integer from a list. given list=[1,'a','c',2,3,4]

[35]:
```python
lst = [1, 'a', 'c', 2, 3, 4]
for item in lst:
    # if isinstance(item, int):
    if type(item) == int:
        print(item)
```

```
1
2
3
4
```

1.2 5. multiplication of each element. given list=[4,5,3,2]

[36]:
```python
lst = [4, 5, 3, 2]
result = 1
for num in lst:
    result *= num
print(result)
```

```
120
```

1.3 6. multiplication table of a given number

[37]:
```python
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

```
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

1.4 7. reverse a list

[38]:
```
lst = [1, 2, 3, 4, 5]
reversed_lst = []
for i in range(len(lst) - 1, -1, -1):
    reversed_lst.append(lst[i])
print(reversed_lst)
```

```
[5, 4, 3, 2, 1]
```

1.5 8. given list is [1,2,3,4] but expected output in new list [2,3,4,5]

[39]:
```
lst = [1, 2, 3, 4]
new_lst = []
for num in lst:
    new_lst.append(num + 1)
print(new_lst)
```

```
[2, 3, 4, 5]
```

1.6 9. Given list is lst=[1,2,3,4] but print 1 and 4 only

[40]:
```
lst = [1, 2, 3, 4]
for i in range(len(lst)):
    if i == 0 or i == len(lst) - 1:
        print(lst[i])
```

```
1
4
```

1.7 11. Given list is [1,2,3,4] but expected output is [1,“a”,2,4]

[41]:
```
given_list = [1, 2, 3, 4]
# modified_list = []
#
# # Iterate through the given list
# for element in given_list:
#     # Check if the element is 3
#     if element == 2:
#         modified_list.append("a")
#     else:
#         modified_list.append(element)
#
# print(modified_list)
```

1.8 12. Given list is [1,2,3,4,5] separate the elements into odd and even categories.

[42]:
```
lst = [1, 2, 3, 4, 5]
evens = []
odds = []
for num in lst:
    if num % 2 == 0:
        evens.append(num)
    else:
        odds.append(num)
print("Evens:", evens)
print("Odds:", odds)
```

```
Evens: [2, 4]
Odds: [1, 3, 5]
```

1.9 13. Given list is [1,2,3,“d”,4,5,“a”] separate the elements based on their data types

[42]:
```

```

1.10 14. Given list is [1,2,3,4,“a”,“b”] append each elements datatypes to separate lists.

[43]:
```
# 14. Given list is [1,2,3,4,"a","b"] append each element's data type to␣
 ↪separate lists
lst = [1, 2, 3, 4, "a", "b"]
int_types = []
str_types = []

for item in lst:
    if isinstance(item, int):
        int_types.append(item)
    elif isinstance(item, str):
        str_types.append(item)

print("Integer elements:", int_types)
print("String elements:", str_types)
```

```
Integer elements: [1, 2, 3, 4]
String elements: ['a', 'b']
```

1.11 15. Python program that accepts a string and calculate the number of digits and letters and space

[44]:
```
# 15. Python program that accepts a string and calculate the number of digits␣
 ↪and letters and space
s = "My name is kamelsh sah"

digits = 0
letters = 0
spaces = 0

for char in s:
    if char.isdigit():
        digits += 1
    elif char.isalpha():
        letters += 1
    elif char.isspace():
        spaces += 1

print("Digits:", digits)
print("Letters:", letters)
print("Spaces:", spaces)
```

```
Digits: 0
Letters: 18
Spaces: 4
```

1.12 16. Python program to check the validity of username and password input by users

[45]:
```
#input from user
username = input("Enter username: ")
password = input("Enter password: ")

# Criteria for validity
valid_username = False
valid_password = False

# Check validity of username
if username.isalnum() and len(username) >= 5:
    valid_username = True

# Check validity of password
if len(password) >= 8:
    has_digit = False
    for char in password:
        if char.isdigit():
```

```
            has_digit = True
            break
    if has_digit:
        valid_password = True

if valid_username and valid_password:
    print("Username and Password are valid")
else:
    if not valid_username:
        print("Invalid Username: Username must be alphanumeric and at least 5␣
 ↪characters long.")
    if not valid_password:
        print("Invalid Password: Password must be at least 8 characters long␣
 ↪and contain at least one digit.")
```

```
Username and Password are valid
```

1.13 17. program to print the given number is odd or even

[46]:
```
num = 19
if num > 0:
    if num % 2 == 0:
        print(f"The number {num} is even")
    else:
        print(f"The number {num} is odd")
else:
    print(f"{num} is a negative number.")
```

```
The number 19 is odd
```

1.14 18. factorial of a given number

[47]:
```
num = 5
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print(factorial)
```

```
120
```

1.15 19. Print multiplication table of 1,2,3,4,5,6,7,8

[48]:
```
for i in range(1, 9):
    for j in range(1, 11):
        print(f"{i} x {j} = {i * j}")
    print()
```

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
```

```
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
```

1.16 20. Given list is lst=[1,2,3,4] but print 1 and 2 only

[49]:
```
lst = [1, 2, 3, 4]
for i in range(len(lst)):
    if i == 0 or i == 1:
        print(lst[i])
```

```
1
2
```

1.17 21. Python program to calculate the sum of all the odd numbers within the given range.

[50]:
```
start = 1
end = 10
odd_sum = 0
for num in range(start, end + 1):
    if num % 2 != 0:
        odd_sum += num
print(odd_sum)
```

```
25
```

1.18 22. Python program to calculate the sum of all the even numbers within the given range.

[51]:
```
start = 1
end = 10
even_sum = 0
for num in range(start, end + 1):
    if num % 2 == 0:
        even_sum += num
print(even_sum)
```

```
30
```

1.19 23. Python program to count the space of a given string

[52]:
```
s = "Python is fun"
space_count = 0
for char in s:
    if char == " ":
        space_count += 1
print(space_count)
```

```
2
```

1.20 24. given list is [1,2,3,4] but expected output is [1,8,27,64]

[53]:
```
lst = [1, 2, 3, 4]
cubed_lst = []
for num in lst:
    cubed_lst.append(num ** 3)
print(cubed_lst)
```

```
[1, 8, 27, 64]
```

1.21 25. reverse of a string a=“programming”.

[54]:
```
s = "programming"
reversed_s = ""
for i in range(len(s) - 1, -1, -1):
    reversed_s += s[i]
print(reversed_s)
```

```
gnimmargorp
```

1.22 26. Place a break statement in the for loop so that it prints from 0 to 7 only (including 7). Given range(50)

[55]:
```
for i in range(50):
    if i > 7:
        break
    print(i)
```

```
0
1
2
3
4
5
6
7
```

1.23 27. Write a for loop that iterates through a string and prints every character.

[56]:
```
s = "softwarica"
for char in s:
    print(char)
```

```
s
o
f
t
w
a
r
i
c
a
```

1.24 28. Write a for loop which print “Hello!,” plus each name in the list. i.e.: “Hello!, ram”. Hint a=[“ram”,“shyam”,1,2] expected output: Hello!ram Hello!shyam

[57]:
```
a = ["ram", "shyam", 1, 2]
for name in a:
    print(f"Hello!, {name}")
```

```
Hello!, ram
Hello!, shyam
Hello!, 1
Hello!, 2
```

1.25 29. Using a for loop and .append() method append each item with a Dr. prefix to the lst. Hint a=[“ram”,“shyam”] expected output: [‘Dr.ram’, ‘Dr.shyam’,‘Dr.1’,‘Dr.2’]

[58]:
```
a = ["ram", "shyam", 1, 2]
dr_lst = []
for name in a:
    dr_lst.append(f"Dr.{name}")
print(dr_lst)
```

```
['Dr.ram', 'Dr.shyam', 'Dr.1', 'Dr.2']
```

1.26 30. Write a for loop which appends the square of each number to the new list.

[59]:
```
lst = [1, 2, 3, 4]
squared_lst = []
for num in lst:
    squared_lst.append(num ** 2)
print(squared_lst)
```

```
[1, 4, 9, 16]
```

1.27 31. Write a for loop using an if statement, that appends each number to the new list if it’s positive. given lst1=[111, 32, -9, -45, -17, 9, 85, -10]

[60]:
```
lst1 = [111, 32, -9, -45, -17, 9, 85, -10]
positive_lst = []
for num in lst1:
    if num > 0:
        positive_lst.append(num)
print(positive_lst)
```

```
[111, 32, 9, 85]
```

1.28 32. Write a Python program that prints all the numbers from 0 to 6 except 3 and 6. given list=[0,1,2,3,4,5,6]

[61]:
```
for num in range(7):
    if num == 3 or num == 6:
        continue
    print(num)
```

```
0
1
2
4
5
```

1.29 33. Write a for loop which appends the type of each element in the first list to the second list.

[62]:
```
first_list = [1, 'a', 3.14, True, None]
second_list = []

for element in first_list:
    second_list.append(type(element).__name__)

print("First list:", first_list)
print("Second list with types:", second_list)
```

```
First list: [1, 'a', 3.14, True, None]
Second list with types: ['int', 'str', 'float', 'bool', 'NoneType']
```

1.30 34. Use else block to display a message “Done” after successful execution of for loop.

[63]:
```
for i in range(5):
    print(i)
else:
    print("Done")
```

```
0
1
2
3
4
Done
```

1.31 35. Write a for loop statement to print the following series:

105 98 ——-7

[64]:
```
for num in range(105, 6, -7):
    print(num)
```

```
105
98
91
84
77
70
63
56
49
42
35
28
21
14
7
```

1.32 36. removal bad characters from the given string. Given bad_chars = [';', ':', '!', "*"], string = "py;th* o:n ! ;py * t*h:o !n". Expected output = pythonpython

[65]:
```
bad_chars = [';', ':', '!', '*', ' ']
string = "py;th* o:n ! ;py * t*h:o In"

# Removing bad characters from the string
for char in bad_chars:
    string = string.replace(char, "")

print(string)
```

```
pythonpythoIn
```

1.33 37. Python program to count the number of even and odd numbers from a series of numbers.

[66]:
```
# Initialize counters for even and odd numbers
even_count = 0
odd_count = 0
# Example series of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Loop through each number in the series
for number in numbers:
    if number % 2 == 0:
        even_count += 1  # Increment even counter
```

```
    else:
        odd_count += 1  # Increment odd counter

# Display the results
print("Number of even numbers:", even_count)
print("Number of odd numbers:", odd_count)
```

```
Number of even numbers: 4
Number of odd numbers: 5
```

1.34 38. Write a python program to display all the prime numbers within a given range.

[67]:
```
start = 10
end = 50

for num in range(start, end + 1):
    if num > 1:
        is_prime = True
        for i in range(2, int(num**0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            print(num)
```

```
11
13
17
19
23
29
31
37
41
43
47
```

1.35 39. given number is prime or not

[68]:
```
num = int(input("Enter a number: "))
for num in range(1, num + 1):
    if num % num == 0 and num % 1 == 0:
        print(num)
```

```
1
2
```

3
4
5
6
7
8
9
10
11
12

1.36 40. program to check given number is palindrome or not

[69]:
```
# Get the number from the user
number = input("Enter a number: ")

# Initialize a variable to store the reverse of the number
reverse_number = ''

# Loop through each digit in the number string in reverse order
for digit in number[::-1]:
    reverse_number += digit

# Check if the reversed number is the same as the original
if number == reverse_number:
    print(number, "is a palindrome.")
else:
    print(number, "is not a palindrome.")
# Get the number from the user
number = input("Enter a number: ")

# Initialize a variable to store the reverse of the number
reverse_number = ''

# Loop through each digit in the number string in reverse order
for digit in number[::-1]:
    reverse_number += digit

# Check if the reversed number is the same as the original
if number == reverse_number:
    print(number, "is a palindrome.")
else:
    print(number, "is not a palindrome.")
```

```
12 is not a palindrome.
10 is not a palindrome.
```

1.37 41. program to check given number is armstrong or not

[70]:
```
# Ask the user for a number
number = int(input("Enter a number: "))

# Convert the number to a string to easily iterate over its digits
num_str = str(number)
num_length = len(num_str)
sum_of_powers = 0

# Iterate over each digit and raise it to the power of the number's length
for digit in num_str:
    sum_of_powers += int(digit) ** num_length

# Compare the sum of powers with the original number
if sum_of_powers == number:
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

```
12 is not an Armstrong number.
```

1.38 42. python program to check a number is perfect number

[71]:
```
# Ask the user for a number
number = int(input("Enter a number to check if it's a perfect number: "))

# Initialize the sum of divisors
sum_of_divisors = 0

# Loop from 1 to number - 1 to find the divisors of the number
for i in range(1, number):
    if number % i == 0:
        sum_of_divisors += i

# Check if the sum of divisors is equal to the number
if sum_of_divisors == number:
    print(f"{number} is a perfect number.")
else:
    print(f"{number} is not a perfect number.")
```

```
10 is not a perfect number.
```

[71]:
```

```