

First__Week

January 9, 2025

1 CS2201 Spring 2025

1.1 Problem Sets

1.1.1 TA Solution by Abhisek Sarkar

as20ms091@iiserkol.ac.in Q1) Lists

- (a) Create a list of integers from 0 to 9 and store the list in variable x
- (b) Create a list of integers from 3 to 12 and store the list in variable y
- (c) Using a single print command, print the list in x in reverse
- (d) Using a single print command, print the list of odd entries (starting from index 1) in x and then the list of even entries (starting from index 0) in x
- (e) Check whether the 4th item of x (0th index element is the 1st element) is same as the 1st item of y by extracting those items
- (f) Print the location of the number 10 in the list x using index()
- (g) Print the location of the number 7 in the list y using index()
- (h) Get a combined (appended) list of the items of x and y
- (i) Create a combined list that consists of x and y. Find the location of the maximum and minimum numbers in this combined list (use max() and min()).

```
[1]: #Q1.a
# list(range(m,n,k)) # generates integers from m to n-1 in intervals of k
x = list(range(0,10,1))
print(x)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
[2]: #Q1.b
y = list(range(3,13))
print(y)
```

[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

```
[3]: #Q1.c
print("Reverse of list x with single command", x[::-1])
```

```

"""
x[::-1]: This slice notation specifies:

    :: Start from the beginning of the list.
    :: Go to the end of the list.
    -1: Step backwards by 1 element at a time.
"""
### Can be done using reverse() function
# x.reverse()
# print(x)

```

Reverse of list x with single command [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

```

[3]: '\nx[::-1]: This slice notation specifies:\n\n    :: Start from the beginning of
the list.\n    :: Go to the end of the list.\n    -1: Step backwards by 1
element at a time.\n'

```

```

[4]: #Q1.d
#odd: Start from 1:
print("The odd numbers in this list", x[1::2])
#even: start from 0
print("The odd numbers in this list", x[::2])

```

The odd numbers in this list [1, 3, 5, 7, 9]

The odd numbers in this list [0, 2, 4, 6, 8]

```

[5]: #Q1.e
if x[3] == y[0]:
    print("the 4th item of x is same as the 1st item of y")
else:
    print("the 4th item of x is not same as the 1st item of y")

```

the 4th item of x is same as the 1st item of y

```

[6]: #Q1.f
if 10 in x:
    print(x.index(10))
else:
    print("x is not in the list")

```

x is not in the list

10 is not in the list stroed in x

```

[7]: #Q1.g
print(y.index(7))

```

4

```
[8]: #Q1.h
# list concatenation (+)
print(x + y)

#using the unpacking operator (*) to unpack the elements of both lists and
↳ create a new list.
z = [*x, *y]
print(z)

# using extend()
x.extend(y)
print(x)
#Caution! it changes the value of x
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

```
[9]: #Q1.i # Location of max and ,im in combined list
print(x.index(max(x)))
print(x.index(min(x)))
```

19

0

Q2) Strings are lists

- Store a string: "The quick brown fox jumps over the lazy dog" in a variable x
- Check whether the word "fox" is in this sentence
- Print the sentence in reverse order
- Print every third character of the above sentence (0th, 3rd, 6th....characters).
- Print every fourth character of the above sentence (0th, 4th, 8th...characters).
- Find how many characters are there in the sentence (including spaces)
- Print every second character of the sentence starting from the last character in reverse order
- Store the first four characters of x in a variable y and the last three letters in a variable z.

Print the output of y + z

- Print the output of y*10

```
[10]: #Q2.a
x = "The quick brown fox jumps over the lazy dog"
print(x)
```

The quick brown fox jumps over the lazy dog

```
[11]: #Q2.b using find(), if the result is -1 then it is not present and if is
      ↪present it will write it's location
      if x.find("fox") != -1 :
          print("The word "fox" is in this sentence")
      else:
          print("The word "fox" is not in this sentence")
```

The word "fox" is in this sentence

```
[12]: # Alternate Solution
      if "fox" in x:
          print("The word "fox" is in this sentence")
      else:
          print("The word "fox" is not in this sentence")
```

The word "fox" is in this sentence

```
[13]: #Q2.c Print the sentence in reverse order
      print(x[::-1])
```

god yzal eht revo spmuj xof nworb kciuq ehT

```
[14]: #Q2.d Print every third character of the above sentence
      print(x[:3])
```

T i o xusv ea g

```
[15]: #Q2.e Print every fourth character of the above sentence
      print(x[:4])
```

Tqkofjsehad

```
[16]: #Q2.f Find how many characters are there in the sentence (including spaces)
      print(len(x))
```

43

```
[17]: #Q2.g Print every second character of the sentence starting from the last
      ↪character in reverse order
      print(x[::-2])
```

gdya h eosmjxfnobkqeT

```
[18]: #Q2.h Store the first four characters of x in a variable y and the last three
      ↪letters in a variable z.

      #Print the output of y + z

      y = x[:4]
      z = x[-3:]
      print(y + z)
```

The dog

x[-3:] is slice notation.

-3 indicates the starting index from the end of the string.

: indicates the ending index (exclusive), which is the end of the string in this case.

Therefore, z will contain the last three characters of the string x.

```
[19]: #Q2,i Print the output of y*10
      print(y*10)
```

The The The The The The The The The

Q.3 Miscellaneous a) Print 'Hello World' on the screen

- b) Add a comment to the above program
- c) Store your name (e.g. Tintin), age (e.g. 20) and roll number (e.g. 20MS1234) in three variables and print them separately and also together like Hello! My name is Tintin . I am 20 years old. My roll number is 20MS1234 using these variables. Use string concatenation.
- d) Take two number strings as input and print their sum
- e) Take two integers as input and print their sum
- f) Use print() to print the statement – It's good to learn Python
- g) Use print() to print the statement – The man asked, "Where to meet you?" I said, "Well, use Google Meet!"
- h) Store an integer, floating point number and character in different variables and print the data type of each variable.
- i) Take your name (e.g. Feluda) in the variable 'Name' as input and print My name is Feluda using Name and string concatenation. Do the same using %s.

```
[20]: #Q3.a
      print("Hello , World!")
```

Hello , World!

```
[21]: #Q.3.b
      print("Hello , World!")
      # This is a comment which is not showing in putput
```

Hello , World!

```
[22]: #Q.3.c
      name = "Abhisek"
      age = "23"
      roll_number = "20MS091"

      # Print Separately
      print(name)
```

```

print(age)
print(roll_number)

# Print together
print("Hello! My name is ", name, "I am", age, "years old. My roll number is",
      ↪roll_number)

```

Abhisek
 23
 20MS091
 Hello! My name is Abhisek I am 23 years old. My roll number is 20MS091

```

[23]: #Q.3.d
#Take two number strings as input and print their sum:
num1_str = input("Enter the first number: ")
num2_str = input("Enter the second number: ")

# Convert strings to integers
num1 = int(num1_str)
num2 = int(num2_str)

# Calculate the sum
sum_of_numbers = num1 + num2

# Print the sum
print("The sum of", num1_str, "and", num2_str, "is:", sum_of_numbers)

```

Enter the first number: 2
 Enter the second number: 3
 The sum of 2 and 3 is: 5

```

[24]: #Q.3.e
# Take two integers as input and print their sum:
num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))

# Calculate the sum
sum_of_integers = num1 + num2

# Print the sum
print("The sum of", num1, "and", num2, "is:", sum_of_integers)

```

Enter the first integer: 5
 Enter the second integer: 6
 The sum of 5 and 6 is: 11

```
[25]: #Q.3.f
# Use print() to print the statement -- It's good to learn Python:
print("It's good to learn Python")
```

It's good to learn Python

```
[26]: #Q.3.g
# Use print() to print the statement -- The man asked, "Where to meet you?" I
↪said, "Well, use Google Meet!"
print("The man asked, \"Where to meet you?\" I said, \"Well, use Google Meet!
↪\"")
```

The man asked, "Where to meet you?" I said, "Well, use Google Meet!"

```
[27]: #Q.3.h
# Store an integer, floating point number and character in different variables
↪and print the data type of each variable.
integer_value = 42
float_value = 3.14
character = 'A'

print(type(integer_value)) # Output: <class 'int'>
print(type(float_value)) # Output: <class 'float'>
print(type(character)) # Output: <class 'str'>
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

```
[28]: #Q.3.i
# Take your name (e.g. Feluda) in the variable 'Name' as input and print My
↪name is Feluda using Name and string concatenation. Do the same using %s.
Name = input("Enter your name: ")

# Using string concatenation
print("My name is", Name)

# Using %s for string formatting
print("My name is %s" % Name)
```

Enter your name: Abhisek

My name is Abhisek

My name is Abhisek

Q4. Write a Python function that takes two lists and returns True if they have at least one common member.

```
[29]: #Q.4
def have_common_member(list1, list2):
```

```

"""
Checks if two lists have at least one common member.

Args:
    list1: The first list.
    list2: The second list.

Returns:
    True if the lists have at least one common member, False otherwise.
"""

for item in list1:
    if item in list2:
        return True
return False

# Example usage:
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
print(have_common_member(list1, list2)) # Output: True

list3 = [1, 2, 3]
list4 = [6, 7, 8]
print(have_common_member(list3, list4)) # Output: False

```

True
False

Q5. Write a Python program to convert a list of characters into a string.

```

[30]: #Q4
x = ['H', 'e', 'l', 'l', 'o']
string_x = "".join(x)
print(string_x) # Output: Hello

```

Hello

Q6. Write a Python program to remove the K'th element from a given list, and print the updated list. Original list: [1, 1, 2, 3, 4, 4, 5, 1] After removing an element at the kth position of the said list: [1, 1, 3, 4, 4, 5, 1]

```

[31]: my_list = [1, 1, 2, 3, 4, 4, 5, 1]

# Using del
del my_list[2] # Remove the element at index 2 (third element)

print("List after removing using del:", my_list)

# Using pop

```



```
my_list = [1, 1, 2, 3, 4, 4, 5, 1]
removed_element = my_list.pop(2) # Remove and return the element at index 2

print("List after removing using pop:", my_list)
print("Removed element:", removed_element)
```

List after removing using del: [1, 1, 3, 4, 4, 5, 1]
List after removing using pop: [1, 1, 3, 4, 4, 5, 1]
Removed element: 2