Open in Colab

(https://colab.research.google.com/github/unt-iialab/INFO5731_Spring2020/blob/master/In_class_exercise/In_class_exercise_01.ipynb)

The first In-class-exercise (08/30/2023, 40 points in total)

This exercise is due tomorrow 08/31/203 11:59 PM.

(1) Write a Python program to calculate the area of a rectangle using user-input length and width.

In [2]:

```
# write your answer here
length = float(input())
width = float(input())
area = length*width
print(area)
```

```
3
2
6.0
```

(2) Create a program that takes two numbers from the user and swaps their values without using a temporary variable.

In [6]:

```python
# write your answer here

a = float(input())
b = float(input())

def swap(a,b):
    a = a+b
    b = a-b
    a = a-b
    return a,b

a,b = swap(a,b)
print(f"after swap : a :{a},b:{b}")
```

```
5
9
after swap : a :9.0,b:5.0
```

(3) Write a Python program to calculate the length of a string.

In [52]:

```python
# write your answer here

s1 = input("enter name:") #"AKhil Chegonda"
lenght_of_string = len(s1)
print("length of string:",lenght_of_string)
```

```
enter name:Sunny
length of string: 5
```

(4) Write a Python program to count the number of characters in a string.

Sample String : 'google.com‘,

Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

In [12]:

```python
# write your answer here

def check_freq(x):
    freq = {}
    for c in set(x):
       freq[c] = x.count(c)
    return freq

check_freq("google.com")
```

Out[12]:

```
{'o': 3, 'g': 2, 'm': 1, 'l': 1, 'e': 1, '.': 1, 'c': 1}
```

(5) Write a Python program to sum all the items in a list.

In [21]:

```python
# write your answer here

list1 = [2,3,4,9]
sum = 0

for i in list1:
    sum += i

print("sum of list",sum)
```

```
sum of list 18
```

(6) Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.

Sample String : 'restart‘,

Expected Result : 'resta$t'

In [23]:

```
# write your answer here
def replace_occurrences(input_string):
    first_char = input_string[0]
    modified_string = first_char + input_string[1:].replace(first_char, '$')
    return modified_string


user_input = input("Enter a string: ")

modified_string = replace_occurrences(user_input)

print(f"The modified string is: {modified_string}")
```

```
Enter a string: restart
The modified string is: resta$t
```

(7) Create a program that calculates the factorial of a number using a while loop.

In [24]:

```
# write your answer here
Given_number = int(input())
fact = 1
a= Given_number

while a >0:
    fact *= a
    a -= 1
print("factorial of a number:",fact)
```

```
5
factorial of a number: 120
```

(8) Create a program that takes a list of numbers as input and outputs a new list with only the even numbers.

In [49]:
```
# write your answer here
list = [1,2,4,5,9,6,15, ]
new=[]
for num in list:
    if num%2==0:
        new.append(num)
        #new=new +[num]

print(new)
```

```
[2, 4, 6]
```

(9) Write a program with python which could accept two parameter a and b to calculate and output the result of S, where

$$S = 3.14 * (1 + a/b)^3$$

In [59]:
```
# write your answer here
a = float(input())
b = float(input())
S = 3.14*((1+(a/b)))**3
print( S)
```

```
2
5
8.616159999999999
```

(10) Write a Python program to concatenate following dictionaries to create a new one.

Sample Dictionary : dic1={1:10, 2:20}, dic2={3:30, 4:40}, dic3={5:50,6:60},

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

In [60]:

```
# write your answer here
# Sample dictionaries
dic1 = {1: 10, 2: 20}
dic2 = {3: 30, 4: 40}
dic3 = {5: 50, 6: 60}

# Create a new dictionary by concatenating the given dictionaries
new_dict = {}
for d in [dic1, dic2, dic3]:
    new_dict.update(d)

print("Concatenated dictionary:", new_dict)
```

```
Concatenated dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

(11) Write a Python function that accepts a tuple of integers and returns the largest and smallest elements.

In [64]:

```
# write your answer here
def find_largest_smallest(numbers_tuple):
    if not numbers_tuple:
        return None, None

    smallest = largest = numbers_tuple[0]

    for num in numbers_tuple:
        if num < smallest:
            smallest = num
        if num > largest:
            largest = num

    return smallest, largest


input_tuple = (3,34,56,8,2,9,24)

smallest, largest = find_largest_smallest(input_tuple)

print(f"Smallest element: {smallest}")
print(f"Largest element: {largest}")
```

```
Smallest element: 2
Largest element: 56
```

(12) Write a Python program that counts the frequency of each character in a given string and stores it in a dictionary.

In [66]:

```
# write your answer here
def count_character_frequency(input_string):
    char_frequency = {}

    for char in input_string:
        if char in char_frequency:
            char_frequency[char] += 1
        else:
            char_frequency[char] = 1

    return char_frequency


user_input = input("Enter a string: ")


frequency_dict = count_character_frequency(user_input)


for char, count in frequency_dict.items():
    print(f"Character '{char}' appears {count} times.")
```

```
Enter a string: university of north texas
Character 'u' appears 1 times.
Character 'n' appears 2 times.
Character 'i' appears 2 times.
Character 'v' appears 1 times.
Character 'e' appears 2 times.
Character 'r' appears 2 times.
Character 's' appears 2 times.
Character 't' appears 3 times.
Character 'y' appears 1 times.
Character ' ' appears 3 times.
Character 'o' appears 2 times.
Character 'f' appears 1 times.
Character 'h' appears 1 times.
Character 'x' appears 1 times.
Character 'a' appears 1 times.
```

(13) Write a Python program to check whether an element exists within a tuple.

In [67]:

```
# write your answer here
def check_element_in_tuple(element, target_tuple):
    return element in target_tuple

example_tuple = (10, 20, 30, 40, 50)


user_input = int(input("Enter an element to check: "))


exists = check_element_in_tuple(user_input, example_tuple)

if exists:
    print(f"The element {user_input} exists in the tuple.")
else:
    print(f"The element {user_input} does not exist in the tuple.")
```

```
Enter an element to check: 25
The element 25 does not exist in the tuple.
```

In [68]:

```
def check_element_in_tuple(element, target_tuple):
    return element in target_tuple

example_tuple = (10, 20, 30, 40, 50)


user_input = int(input("Enter an element to check: "))


exists = check_element_in_tuple(user_input, example_tuple)

if exists:
    print(f"The element {user_input} exists in the tuple.")
else:
    print(f"The element {user_input} does not exist in the tuple.")
```

```
Enter an element to check: 20
The element 20 exists in the tuple.
```

(14) Write a Python program to find maximum and the minimum value in a set.

In [70]:

```python
# write your answer here
def find_max_min_in_set(input_set):
    if not input_set:
        return None, None

    max_value = max(input_set)
    min_value = min(input_set)
    return max_value, min_value


example_set = {15, 30, 10, 45, 5}

max_value, min_value = find_max_min_in_set(example_set)

print(f"Maximum value in the set: {max_value}")
print(f"Minimum value in the set: {min_value}")
```

```
Maximum value in the set: 45
Minimum value in the set: 5
```

(15) Write a Python program to randomly divide the students in this class into six groups for term projects, each group should have at least 4 students but no more than 6 students (including 6). Here is the students list: student_list_fall_2023=['sb1826' , 'lb0582' , 'ab1820' , 'sb1883' , 'mb1296' , 'sc1622' , 'tc0469' , 'ac1770' , 'mc1453' , 'ad1187' , 'sd0926' , 'sd1268' , 'ag1609' , 'sg1638' , 'mg1299' , 'vkk0022' , 'bk0413' , 'gk0257' , 'am2807' , 'gm0558' , 'pm0658' , 'am2346' , 'rm1090' , 'sm2364' , 'sm2747' , 'sm2343' , 'vm0597' , 'htn0086' , 'sp2075' , 'gp0306' , 'lp0502' , 'sr1508' , 'vr0331' , 'yr0180' , 'as2819' , 'ss2981' , 'ns0756' , 'mt0825' , 'dt0463' , 'su0137' , 'dv0310']

In [94]:

```python
# write your answer here
import random

# List of students
student_list_fall_2023 = ['sb1826', 'lb0582', 'ab1820', 'sb1883', 'mb1296', 'sc1622',
                          'tc0469', 'ac1770', 'mc1453', 'ad1187', 'sd0926', 'sd1268',
                          'ag1609', 'sg1638', 'mg1299', 'vkk0022', 'bk0413', 'gk0257',
                          'am2807', 'gm0558', 'pm0658', 'am2346', 'rm1090', 'sm2364',
                          'sm2747', 'sm2343', 'vm0597', 'htn0086', 'sp2075', 'gp0306',
                          'lp0502', 'sr1508', 'vr0331', 'yr0180', 'as2819', 'ss2981',
                          'ns0756', 'mt0825', 'dt0463', 'su0137', 'dv0310']

# Shuffle the student list
random.shuffle(student_list_fall_2023)

groups = [student_list_fall_2023[i:i+4] for i in range(0, len(student_list_fall_2023), 4)]

# Display the groups
for i, group in enumerate(groups, start=1):
    print(f"Group {i}: {group}")
```

```
Group 1: ['ab1820', 'bk0413', 'sm2747', 'vkk0022']
Group 2: ['dv0310', 'htn0086', 'sd0926', 'ac1770']
Group 3: ['lp0502', 'sc1622', 'ns0756', 'vr0331']
Group 4: ['sg1638', 'vm0597', 'mc1453', 'ad1187']
Group 5: ['gm0558', 'su0137', 'yr0180', 'dt0463']
Group 6: ['sm2364', 'am2346', 'ss2981', 'sr1508']
Group 7: ['gp0306', 'sp2075', 'mb1296', 'sd1268']
Group 8: ['mg1299', 'as2819', 'sb1883', 'rm1090']
Group 9: ['sb1826', 'mt0825', 'pm0658', 'tc0469']
Group 10: ['lb0582', 'ag1609', 'gk0257', 'am2807']
Group 11: ['sm2343']
```

(16) Write a Python program to find the duplicate elements in a given array of integers. Return -1 If there are no such elements.

In [76]:

```python
# write your answer here
# Get input from the user
user_input = input("Enter integers separated by spaces: ")
input_array = [int(num) for num in user_input.split()]

duplicate_set = set()
unique_set = set()

for num in input_array:
    if num in unique_set:
        duplicate_set.add(num)
    else:
        unique_set.add(num)

if not duplicate_set:
    print("No duplicate elements found.")
else:
    print("Duplicate elements:", duplicate_set)
```

```
Enter integers separated by spaces: 2 3 3 4 5 1 2 2 5
Duplicate elements: {2, 3, 5}
```

(17) Write a Python program to select all the Sundays of a specified year.

In [78]:

```
# write your answer here
import calendar

def find_sundays(year):
    sundays = []

    for month in range(1, 13):
        for day in range(1, calendar.monthrange(year, month)[1] + 1):
            if calendar.weekday(year, month, day) == calendar.SUNDAY:
                sundays.append((year, month, day))

    return sundays


year = int(input("Enter a year: "))

sundays = find_sundays(year)
if sundays:
    for sunday in sundays:
        print(f"Sunday: {sunday[0]}-{sunday[1]:02d}-{sunday[2]:02d}")
else:
    print("No Sundays found in the specified year.")
```

```
Enter a year: 2023
Sunday: 2023-01-01
Sunday: 2023-01-08
Sunday: 2023-01-15
Sunday: 2023-01-22
Sunday: 2023-01-29
Sunday: 2023-02-05
Sunday: 2023-02-12
Sunday: 2023-02-19
Sunday: 2023-02-26
Sunday: 2023-03-05
Sunday: 2023-03-12
Sunday: 2023-03-19
Sunday: 2023-03-26
Sunday: 2023-04-02
Sunday: 2023-04-09
Sunday: 2023-04-16
Sunday: 2023-04-23
Sunday: 2023-04-30
Sunday: 2023-05-07
Sunday: 2023-05-14
Sunday: 2023-05-21
Sunday: 2023-05-28
Sunday: 2023-06-04
Sunday: 2023-06-11
Sunday: 2023-06-18
Sunday: 2023-06-25
Sunday: 2023-07-02
Sunday: 2023-07-09
Sunday: 2023-07-16
Sunday: 2023-07-23
Sunday: 2023-07-30
Sunday: 2023-08-06
Sunday: 2023-08-13
Sunday: 2023-08-20
Sunday: 2023-08-27
Sunday: 2023-09-03
Sunday: 2023-09-10
Sunday: 2023-09-17
Sunday: 2023-09-24
Sunday: 2023-10-01
Sunday: 2023-10-08
Sunday: 2023-10-15
```

```
Sunday: 2023-10-22
Sunday: 2023-10-29
Sunday: 2023-11-05
Sunday: 2023-11-12
Sunday: 2023-11-19
Sunday: 2023-11-26
Sunday: 2023-12-03
Sunday: 2023-12-10
Sunday: 2023-12-17
Sunday: 2023-12-24
Sunday: 2023-12-31
```

(18) Build a program that calculates the factorial of a given positive integer. The factorial of a non-negative integer n is the product of all positive integers less than or equal to n.

In [84]:

```
# write your answer here
def calculate_factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * calculate_factorial(n - 1)

n = int(input("Enter a positive integer: "))

if n < 0:
    print("Factorial is not defined for negative integers.")
else:
    factorial = calculate_factorial(n)
    print(f"The factorial of {n} is: {factorial}")
```

```
Enter a positive integer: 10
The factorial of 10 is: 3628800
```

(19) Write a program that takes a positive integer n as input and prints the count of even numbers from 1 to n. Use a loop to iterate through the numbers and utilize the break statement to optimize the loop.

In [81]:

```
# write your answer here
def count_even_numbers(n):
    count = 0
    for num in range(1, n + 1):
        if num % 2 == 0:
            count += 1
        elif num > n:  # Optimization using break
            break
    return count


n = int(input("Enter a positive integer: "))

if n <= 0:
    print("Please enter a positive integer.")
else:
    even_count = count_even_numbers(n)
    print(f"The count of even numbers from 1 to {n} is: {even_count}")
```

```
Enter a positive integer: 25
The count of even numbers from 1 to 25 is: 12
```

(20) Write a Python function called count_vowels that takes a string as input and returns the count of vowels (both lowercase and uppercase) present in the string. Consider the vowels to be 'a', 'e', 'i', 'o', and 'u'. Ignore spaces and other non-alphabetic characters.

In [82]:

```
# write your answer here
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0

    for char in input_string:
        if char in vowels:
            vowel_count += 1

    return vowel_count

user_input = input("Enter a string: ")

vowel_count = count_vowels(user_input)
print(f"The count of vowels in the string is: {vowel_count}")
```

```
Enter a string: kaushal sen
The count of vowels in the string is: 4
```