

(Q1) Write a Python program to calculate the area of a rectangle given its length and width?

A Python program to calculate the area of a rectangle given its length and width

```
l = int(input('Enter Length of the Rectangle: '))
b = int(input('Enter Breadth of the Rectangle: '))
print(f" Area of rectangle is: {(l*b)}")
```

(Q2) Write a program to convert miles to kilometers?

A simple Python program to convert miles to kilometers

```
miles = int(input("Please enter miles:"))
print(miles*1.6, " Kms")
```

(Q3) Write a function to check if a given string is a palindrome?

```
def isPalindrome(str):
    if (str == str[::-1]) :
        return "The string is a palindrome."
    else:
        return "The string is not a palindrome."
str = input ("Enter string: ")
print(isPalindrome(str))
```

This function **is_palindrome()** takes a string as input, removes spaces and converts it to lowercase, then checks if the string is equal to its reverse (using slicing **[::-1]**). If they are equal, it returns **True**, indicating that the string is a palindrome; otherwise, it returns **False**.

(Q4) Write a Python program to find the second largest element in a list?

This program defines a function **second_largest()** that takes a list as input and iterates through it to find the second largest element.

It initializes **first** and **second** to negative infinity. Then, it traverses the list and updates **first** and **second** accordingly. Finally, it returns the value of the second largest element found.

```
i = input("Enter elements of the list separated by spaces: ")
li= list(map(int, i.split()))
sort = sorted(set(li))
if len(sort) >= 2:
```

```
print("Second largest element is:", sort[-2])  
else:  
print("No second largest element found.")
```

(Q5) Explain what indentation means in Python?

1. Indentation can be achieved by four spaces or tab space in workspace
2. It helps Python determine the structure of the code
3. It helps Python determine the scope of various constructs like loops, conditionals, and function definitions.

Effects:

- Incorrect indentation can lead to
1. syntax errors
 2. change the logical structure of the code
 3. potentially leading to unintended behaviour.

Example:

```
If(age>=18):  
Print("Eligible to vote")  
else:  
Print("Not Eligible to vote")
```

(The print statement outside the if block is not indented, indicating that it is not part of the if block.)

(Q6) Write a program to perform set difference operation?

In this program, the `set_difference()` function takes two sets as input and calculates the difference between them using the `-` operator

Alternatively, you can use the `difference()` method commented out in the function. Finally, the program prints the result of the set difference operation

```
def user_input():  
    i= input("Enter elements of the set separated by spaces: ")  
    return set(map(int, i.split()))  
  
set1 = user_input()
```

```

set2 = user_input()
print("Union:", set1 | set2 )
print("Intersection:", set1 & set2)
print("Difference:", set1 - set2)
print("Symmetric Difference:", set1 ^ set2)
print("Is Subset:", set1 <= set2)
print("Is Superset:", set1 >= set2)
print("Are Disjoint:", set1.isdisjoint(set2))

```

(Q7) Write a Python program to print numbers from 1 to 10 using a while loop?

This program initializes a variable `num` with the value 1 and enters a while loop that continues as long as `num` is less than or equal to 10.

Inside the loop, the current value of `num` is printed, and then `num` is incremented by 1 using the `+=` operator

This process repeats until `num` reaches 10, at which point the loop terminates.

```

n= 1
while n <= 10:
    print(n)
    n+=1

```

(Q8) Write a program to calculate the factorial of a number using a while loop?

The `factorial()` function takes an integer `n` as input and calculates its factorial
 If `n` is negative, the function returns a message indicating that factorial is not defined for negative numbers

If `n` is 0 or 1, the function returns 1 because the factorial of 0 and 1 is 1

Otherwise, the function calculates the factorial using a while loop. It initializes `factorial_result` to 1 and multiplies it by `n`, then decrements `n` by 1 in each iteration until `n` becomes 1

Finally, it returns the calculated factorial result

```

i = int(input("Enter a non-negative integer: "))
if i < 0:
    print("Factorial is not defined for negative numbers.")
elif i == 0 or i == 1:
    print("The factorial of", i, "is: 1")

```

```

else
    result = 1
    n = i
    while n > 1:
        result *= n
        n -= 1
    print(f"The factorial of {i} is: {result}")

```

(Q9) Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements?

If the number is greater than 0, it prints "The number is positive"

If the number is less than 0, it prints "The number is negative"

If the number is neither greater than nor less than 0, it prints "The number is zero"

```
n = int(input("Enter a number: "))
```

```
if n > 0:
```

```
    print("The number is positive.")
```

```
elif n < 0:
```

```
    print("The number is negative.")
```

```
else:
```

```
    print("The number is zero.")
```

(Q10) Write a program to determine the largest among three numbers using conditional Statements?

If **num1** is greater than or equal to **num2** and **num1** is greater than or equal to **num3**, then **num1** is the largest

If **num2** is greater than or equal to **num1** and **num2** is greater than or equal to **num3**, then **num2** is the largest

If neither of the above conditions is true, then **num3** is the largest

```
n1 = int(input("Enter the first number: "))
```

```
n2 = int(input("Enter the second number: "))
```

```
n3 = int(input("Enter the third number: "))
```

```
l = n1
```

```
if n2 > l:
```

```

l = num2
if n3 > l:
    l = num3
print("Largest number is:", l)

```

(Q11) Write a Python program to create a numpy array filled with ones of given shape?

This program imports the NumPy library (**import numpy as np**) and defines a function **ones_array()** that takes the shape of the desired array as input. Inside the function, it uses **np.ones(shape)** to create a NumPy array filled with ones of the given shape

You can run this program, enter the shape of the array when prompted (e.g., for a 2D array, you can input two integers separated by space like **3 4**), and it will generate and display the array filled with ones of the specified shape

```

import numpy as np

i = input("Enter the shape of the array separated by spaces : ")
print(np.ones(tuple(map(int, i.split()))))

#i- input by user is "3 4".
#split()- returns a list of substrings. it returns ["3", "4"].
#map()-converts each string in the list to an integer. It becomes [3, 4].
#tuple()-converts the map object returned by map() into a tuple.(3,4).

```

(Q12) Write a program to create a 2D numpy array initialized with random integers?

```

import numpy as np

rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))
start = int(input("Enter the begin value of range: "))
stop = int(input("Enter the last value of range: "))
print(np.random.randint(strat,stop,(rows, cols)))

```

(Q13) Write a Python program to generate an array of evenly spaced numbers over a specified range using linspace?

We import the NumPy library (**import numpy as np**)

We define a function **generate_linspace()** that takes the start value, stop value, and the number of elements as input

Inside the function, we use `np.linspace(start, stop, num)` to generate an array of `num` evenly spaced numbers over the range from `start` to `stop`, inclusive

Syntax: `np.linspace(start,end,number of elements)`-evenly spaced step size

```
import numpy as np
```

```
start = int(input("Enter the start of the range: "))
```

```
end = int(input("Enter the end of the range: "))
```

```
num= int(input("Enter the number of points: "))
```

```
print(np.linspace(start, end, num))
```

(Q14) Write a program to generate an array of 10 equally spaced values between 1 and 100 using Linspace?

We import the NumPy library (`import numpy as np`)

We use `np.linspace(1, 100, 10)` to generate an array of 10 equally spaced values between 1 and 100, inclusive

The `linspace()` function takes three arguments: the start value (`1`), the stop value (`100`), and the number of elements (`10`)

```
import numpy as np
```

```
print(np.linspace(1, 100, 10))
```

(Q15) Write a Python program to create an array containing even numbers from 2 to 20 using Arange?

Syntax: `np.arange(start,end,stepsize)`-number of elements will depend on stepsize and end

value

Note: end value should be exceed up on one by our required value

```
import numpy as np
```

```
print(np.arange(2, 21, 2))
```

(Q16) Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5?

We import the NumPy library (`import numpy as np`)

We use `np.arange(1, 11, 0.5)` to create an array containing numbers from 1 to 10 with a step size of 0.5

The `arange()` function takes three arguments: the start value (**1**), the stop value (**11**), and the step size (**0.5**). The stop value is exclusive, so the sequence stops before reaching 11

using `arange`.

```
import numpy as np
```

```
print(np.arange(1, 11, 0.5))
```