

ASSIGNMENT 1 ANSWERS

DATE: 14-FEB-2024

1. Write a Python program to calculate the area of a rectangle given its length and width.

A)



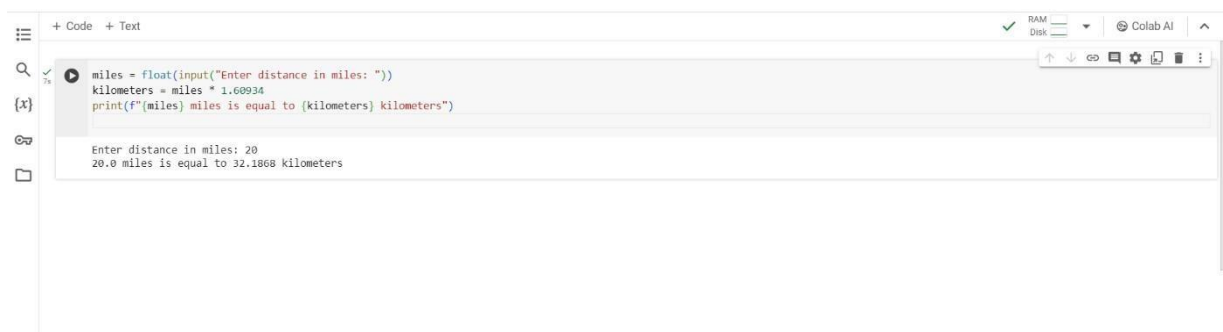
```
+ Code + Text
length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

area = length * width
print(f"The area of the rectangle is: {area}")
```

Enter the length of the rectangle: 5
Enter the width of the rectangle: 4
The area of the rectangle is: 20.0

2. Write a program to convert miles to kilometres

A)

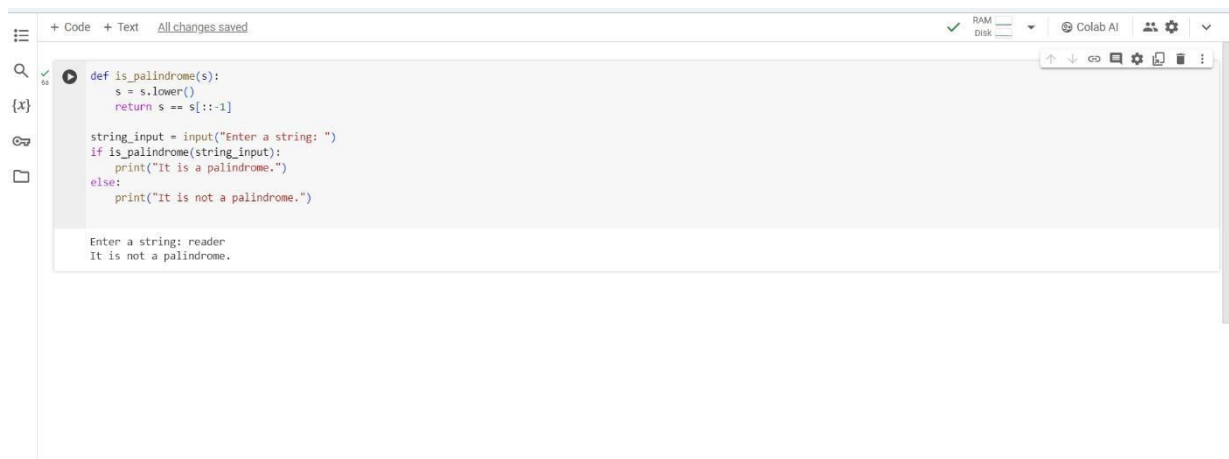


```
+ Code + Text
miles = float(input("Enter distance in miles: "))
kilometers = miles * 1.60934
print(f"{miles} miles is equal to {kilometers} kilometers")
```

Enter distance in miles: 20
20.0 miles is equal to 32.1868 Kilometers

3. Write a function to check if a given string is a palindrome.

A)



```
+ Code + Text All changes saved
def is_palindrome(s):
    s = s.lower()
    return s == s[::-1]

string_input = input("Enter a string: ")
if is_palindrome(string_input):
    print("It is a palindrome.")
else:
    print("It is not a palindrome.")
```

Enter a string: reader
It is not a palindrome.

4. Write a Python program to find the second largest element in a list.

A)



```
my_list = [int(x) for x in input("Enter elements of the list separated by space: ").split()]
sorted_list = sorted(set(my_list), reverse=True)
second_largest = sorted_list[1]
print(f"The second largest element is: {second_largest}")
```

Enter elements of the list separated by space: 16 54 58 45
The second largest element is: 54

5) Explain what indentation means in Python.

A)

In Python, indentation is a way of indicating the grouping of statements within a block of code. It is used to define the structure and scope of the code. Instead of using curly braces or keywords like "begin" and "end" as in some other programming languages, Python uses indentation to show which statements belong to a particular block of code.

Indentation is typically done using spaces or tabs at the beginning of lines. Consistent indentation is crucial because it determines the code's structure. Statements with the same level of indentation are considered part of the same block.

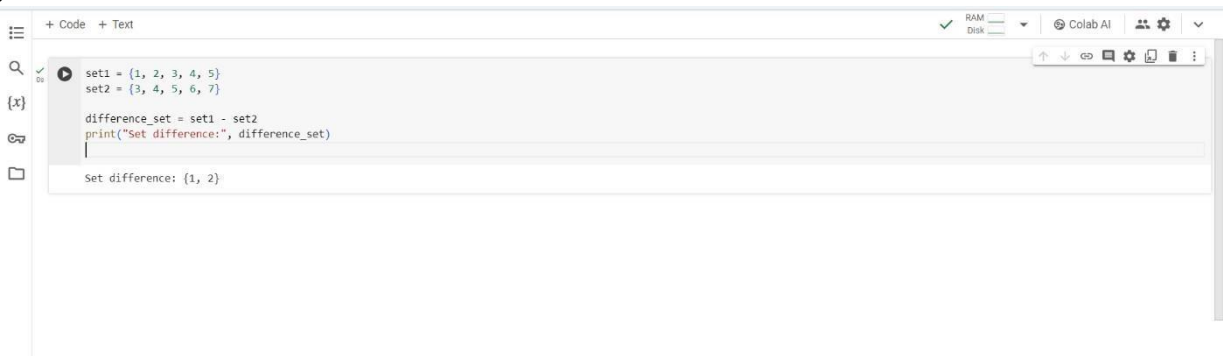
For example:

```
if x > 0:
    print("x is positive")
    print("This statement is also inside
the if block") # Statements without indentation are outside
the block
print("This statement is outside the if block")
```

In the above example, the two **print** statements are inside the **if** block because they are indented under it. The last **print** statement is not indented and is outside the **if** block.

6) Write a program to perform set difference operation.

A)

A screenshot of a Google Colab notebook. The code cell contains two sets, set1 = {1, 2, 3, 4, 5} and set2 = {3, 4, 5, 6, 7}. It calculates the difference set as set1 - set2 and prints the result. The output cell shows 'Set difference: {1, 2}'.

```
+ Code + Text
```

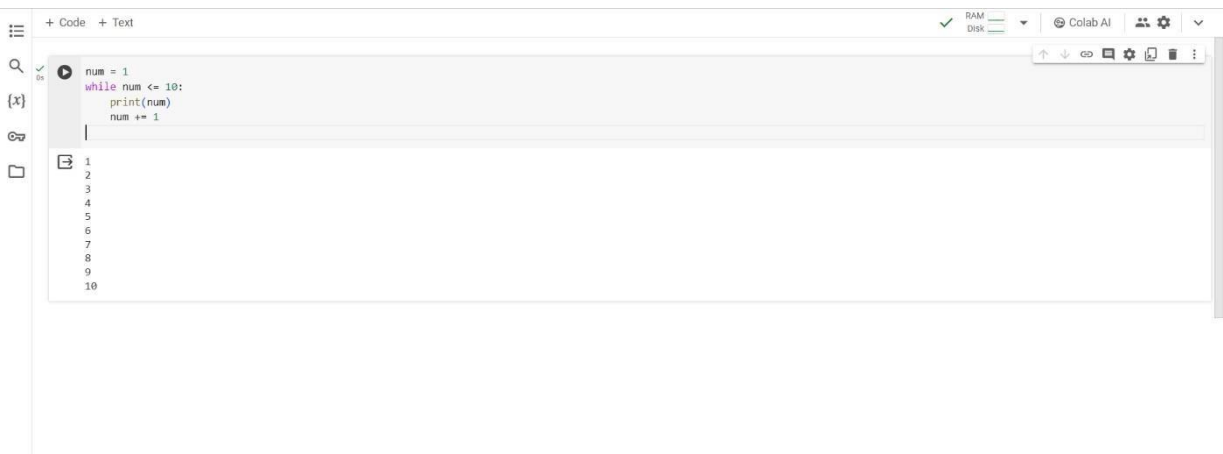
```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}

difference_set = set1 - set2
print("Set difference:", difference_set)
```

```
Set difference: {1, 2}
```

7. Write a Python program to print numbers from 1 to 10 using a while loop.

A)

A screenshot of a Google Colab notebook. The code cell initializes num = 1 and enters a while loop that prints num and increments it by 1 until num is 10. The output cell shows the numbers 1 through 10 printed on separate lines.

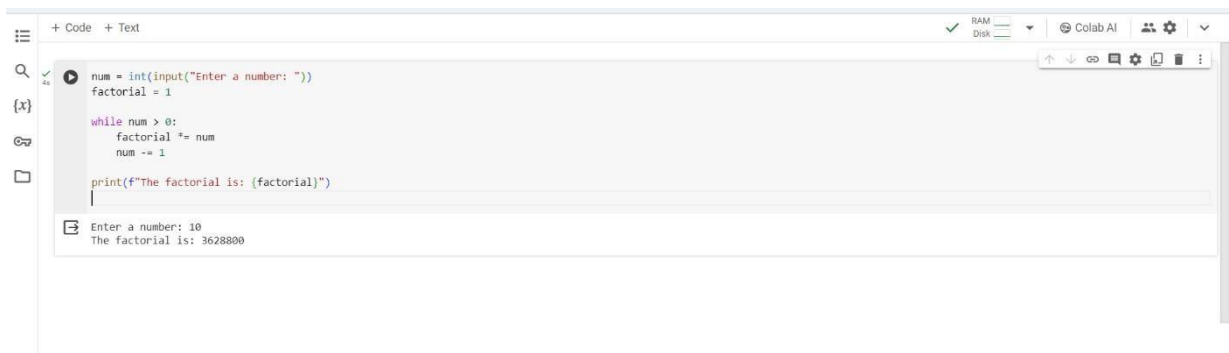
```
+ Code + Text
```

```
num = 1
while num <= 10:
    print(num)
    num += 1
```

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

8. Write a program to calculate the factorial of a number using a while loop.

A)

A screenshot of a Google Colab notebook. The code cell takes user input for a number, initializes factorial = 1, and uses a while loop to calculate the factorial. The output cell shows the input '10' and the resulting factorial '3628800'.

```
+ Code + Text
```

```
num = int(input("Enter a number: "))
factorial = 1

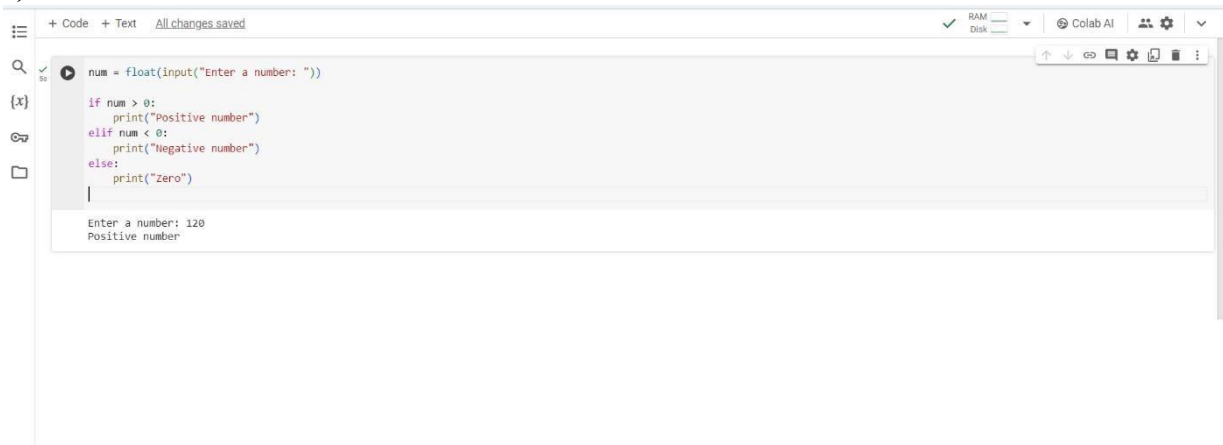
while num > 0:
    factorial *= num
    num -= 1

print(f"The factorial is: {factorial}")
```

```
Enter a number: 10
The factorial is: 3628800
```

9. Write a Python program to check if a number is positive, negative, or zero using ifelifelse statement

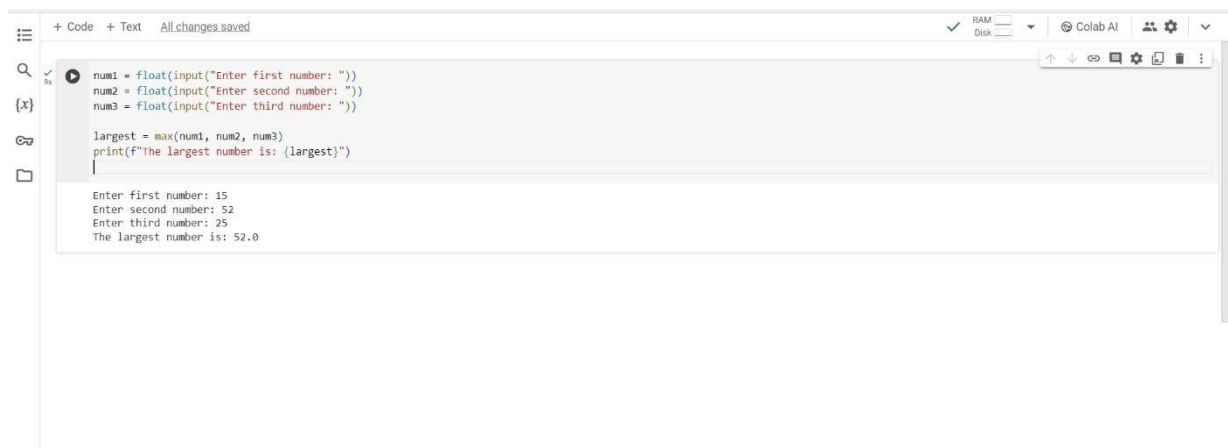
A)



```
+ Code + Text All changes saved
num = float(input("Enter a number: "))
if num > 0:
    print("Positive number")
elif num < 0:
    print("Negative number")
else:
    print("Zero")
|
Enter a number: 120
Positive number
```

10. Write a program to determine the largest among three numbers using conditional statements.

A)

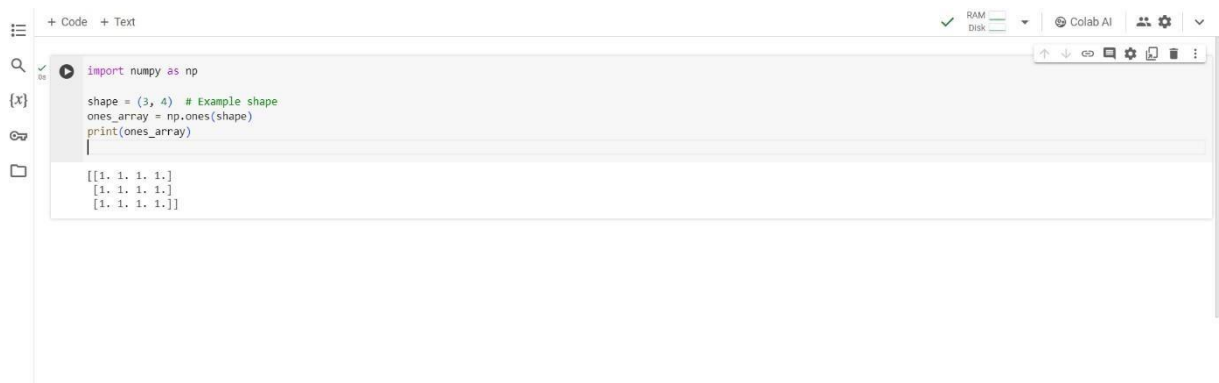


```
+ Code + Text All changes saved
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

largest = max(num1, num2, num3)
print(f"The largest number is: {largest}")
|
Enter first number: 15
Enter second number: 52
Enter third number: 25
The largest number is: 52.0
```

11. Write a Python program to create a numpy array filled with ones of given shape.

A)

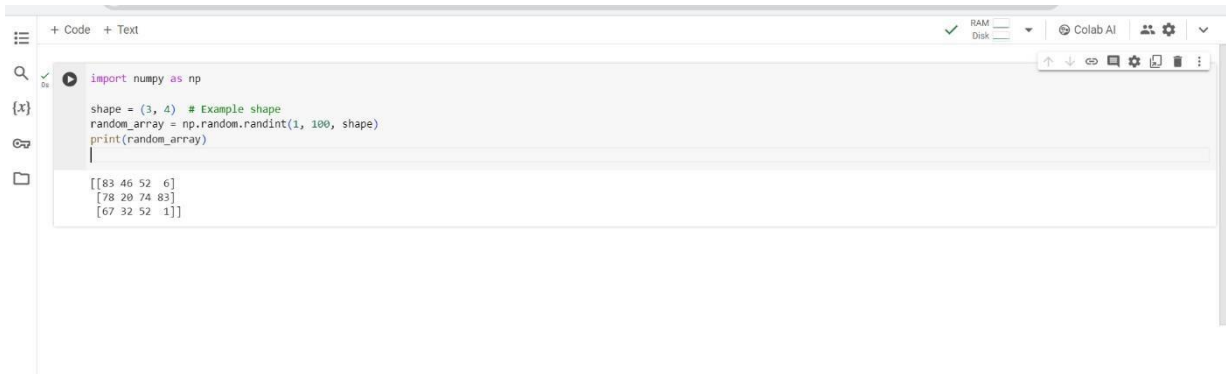


```
+ Code + Text
import numpy as np

shape = (3, 4) # Example shape
ones_array = np.ones(shape)
print(ones_array)
|
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
```

12. Write a program to create a 2D numpy array initialized with random integers.

A)



```
+ Code + Text
```

```
import numpy as np

shape = (3, 4) # Example shape
random_array = np.random.randint(1, 100, shape)
print(random_array)
|
```

```
[[83 46 52 6]
 [78 20 74 83]
 [67 32 52 1]]
```

13. Write a Python program to generate an array of evenly spaced numbers over a specified range using linspace.

A)



```
+ Code + Text All changes saved
```

```
import numpy as np

start = 1
end = 5
num_points = 10
evenly_spaced_array = np.linspace(start, end, num_points)
print(evenly_spaced_array)
|
```

```
[1.         1.44444444 1.88888889 2.33333333 2.77777778 3.22222222
 3.66666667 4.11111111 4.55555556 5.]
```

14. Write a program to generate an array of 10 equally spaced values between 1 and 100 using linspace.

A)



```
+ Code + Text
```

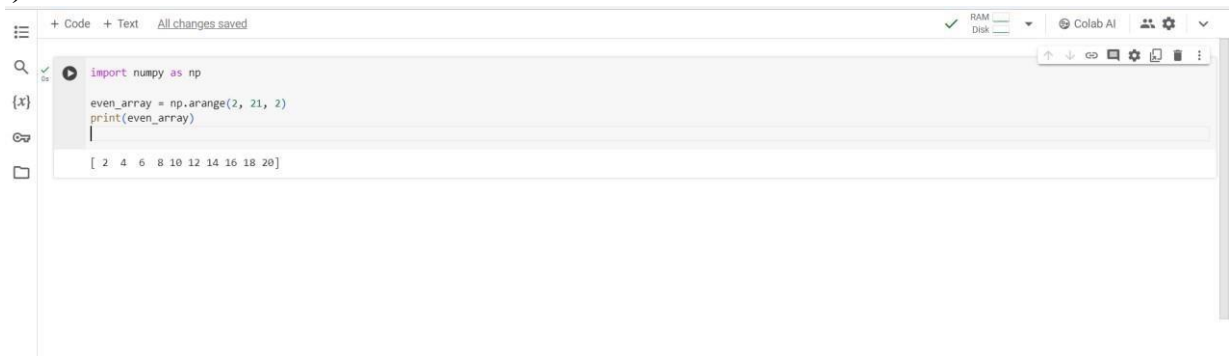
```
import numpy as np

start = 1
end = 100
num_points = 10
equally_spaced_array = np.linspace(start, end, num_points)
print(equally_spaced_array)
|
```

```
[ 1.  12.  23.  34.  45.  56.  67.  78.  89. 100.]
```

15. Write a Python program to create an array containing even numbers from 2 to 20 using arange.

A)



```
+ Code + Text All changes saved
```

```
import numpy as np

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

```
[ 2  4  6  8 10 12 14 16 18 20]
```

16. Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using arange

A)



```
+ Code + Text All changes saved
```

```
import numpy as np

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

```
[ 1.  1.5  2.  2.5  3.  3.5  4.  4.5  5.  5.5  6.  6.5  7.  7.5
  8.  8.5  9.  9.5 10. 10.5]
```