Assessment-01

AI/ML

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

AREA OF RECTANGLE

In [4]:
```
def areaOfRect(x, y):
    return x*y
print("Enter length of Rectangle:")
print("Enter breadth of Rectangle:")
l = float(input())
b = float(input())
a = areaOfRect(l, b)
print("\n Area= ",a)
```

```
Enter length of Rectangle:
Enter breadth of Rectangle:
2
4

 Area=  8.0
```

2. Write a program to convert miles to kilometers.

CONVERT MILES TO KILOMETERS

In [5]:
```
miles = float(input("Enter the value in miles:"))
conversion_factor=1.6
kilometers = miles* conversion_factor
print("kilometers= ",kilometers)
```

```
Enter the value in miles:5
kilometers=  8.0
```

In [ ]:

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

palindrome

```
In [7]: def isPalindrone(s):
            return s == s[::-1]
        s = "madam"
        ans = isPalindrone(s)
        if ans:
            print("yes")
        else:
            print("no")
```

```
yes
```

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

second largest number in list

```
In [8]: a=[]
        n=int(input("Enter number of elements:"))
        for i in range (1,n+1):
            b=int(input("Enter element:"))
            a.append(b)
        a.sort()
        print("second largest element is:",a[n-2])
```

```
Enter number of elements:5
Enter element:12
Enter element:23
Enter element:56
Enter element:78
Enter element:45
second largest element is: 56
```

5.Explain what indentation means in Python.

In Python, indentation is used to define the structure and hierarchy of the code. Unlike many other programming languages

that use braces or keywords to indicate blocks of code, Python relies on consistent indentation to determine the grouping of statements.

Indentation is crucial in Python because it helps in visually organizing code and indicating the scope of control structures (such as loops, conditionals, and functions). The standard convention in Python is to use four spaces for each level of indentation.

6. Write a program to perform set difference operation.

set difference operation

```
In [9]: A={1,3,4,5,6}
        B={2,4,5,1,9}
        print(A.difference(B))
        print(B.difference(A))
```

```
{3, 6}
{9, 2}
```

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

naturals numbers using while

```
In [10]: i=1
         while i<11:
             print (i)
             i=i+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.

Factorial number using while

In [13]:
```
print("Enter the number:")
num = int(input())
fact = 1
i = 1
while i<=num:
    fact = fact* i
    i = i+1
print("\n Factorial=",fact)
```

```
Enter the number:
6

 Factorial= 720
```

9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements.

Checking if a number is +,- or 0

In [14]:
```
num = float(input("Enter a number:"))
if num>0:
     print("Positive")
elif num==0:
     print("Zero")
else:
    print("Negative")
```

```
Enter a number:-9
Negative
```

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

Largest of the three numbers

In [15]:
```
a = 5
b = 6
c = 7
largest = 0
if a>b and a>c:
    largest = a
if b>a and b>c:
    largest = b
if c>a and c>b:
    largest = c
print(largest,"is the largest of the  number")
```

```
7 is the largest of the  number
```

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

Creating a numpy array

In [16]:
```
import numpy as np
```

In [17]:
```
a = np.ones((2,3))
print(a)
```

```
[[1. 1. 1.]
 [1. 1. 1.]]
```

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

2D numpy array

```
In [18]: rows, cols = 2,3
         random_array = np.random.randint(1,10, size=(rows, cols))
         print("2D Array with random integers:")
         print(random_array)
```

```
2D Array with random integers:
[[3 3 3]
 [6 8 1]]
```

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

Array of evenly spaced numbers

```
In [19]: def generate_evenly_spaced_array(start, end, num_points):

             evenly_spaced_array = np.linspace(start, end, num_points)
             return evenly_spaced_array

         # Example usage:
         start_range = 0
         end_range = 10
         num_points = 5

         result_array = generate_evenly_spaced_array(start_range, end_range, num_points)
         print("Generated Array:", result_array)
```

```
Generated Array: [ 0.   2.5  5.   7.5 10. ]
```

14. Write a program to generate an array of 10 equally spaced

values between 1 and 100 using linspace.

Array of 10 equally spaced values

```
In [21]: result_array = np.linspace(1, 100, 10)
         print("Generated Array:", result_array)
```

```
Generated 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 arrange.

Creating an array containing even numbers

```
In [22]: even_numbers_array = np.arange(2, 21, 2)
         print("Array of Even Numbers:", even_numbers_array)
```

```
Array of Even Numbers: [ 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 arrange.

Creating an array containing numbers from 1 to 10

```
In [23]: array_with_step = np.arange(1, 10.5, 0.5)
         print("Array with Step Size 0.5:", array_with_step)
```

```
Array with Step Size 0.5: [ 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. ]
```