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

In [78]:
```
def area( l, w ):
    return l*w

length, width = map( int, input().split() )
res = area( length, width )
print( res )
```

```
2 3
6
```

2. Write a program to convert miles to kilometers.

In [79]:
```
# 1 mile = 1.60934 kilometers
def miles_to_kilometers( miles ):
    return miles * 1.60934

miles = float( input() )
res = miles_to_kilometers( miles )
print( res )
```

```
2
3.21868
```

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

In [80]:
```
def check_if_palindrome_or_not( string ):
    return string == string[::-1]

string = input()
res = check_if_palindrome_or_not( string )
print( res )
```

```
racecar
True
```

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

In [81]:
```
def second_largest( array ):
    array.sort()
    return array[-2]

array = list( map( int, input().split() ) )
res = second_largest( array )
print( res )
```

```
9 7 6 3 8 4 2 4
8
```

5. Explain what indentation means in Python.

In Python, indentation refers to the whitespace (spaces or tabs) that precedes lines of code within blocks, such as loops, conditional statements, and function definitions. Indentation is used to denote the structure and nesting of code blocks, rather than using curly braces or keywords like "begin" and "end" as in other programming languages.

6. Write a program to perform set difference operation

In [82]:
```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}

res = set1 - set2
print( res )
```

```
{1, 2}
```

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

In [83]:
```
i = 0
while( i < 10 ):
    print( i:= i + 1, end = ' ' )
```

```
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

In [84]:
```
def factorial(num):
    res = 1
    while num > 1:
        res *= num
        num -= 1
    return res

num = int(input())
res = factorial(num)
print(res)
```

```
5
120
```

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

In [85]:
```
num = int( input() )
if num > 0:
    print( "positive" )
elif num == 0:
    print( "zero" )
else:
    print( "negative" )
```

```
2
positive
```

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

In [86]:
```
a, b, c = map( int, input().split() )
if a >= b and a >= c:
    print( a )
elif b >= c:
    print( b )
else:
    print( c )
```

```
2 3 1
3
```

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

In [87]:
```
import numpy as np

rows, cols = map( int, input().split() )
shape = ( rows, cols )
res = np.ones( shape )
print( res )
```

```
2 3
[[1. 1. 1.]
 [1. 1. 1.]]
```

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

In [88]:
```
import numpy as np

rows, cols = map( int, input().split() )
low, high = map( int, input().split() )
shape = ( rows, cols )
res = np.random.randint( low, high, size= shape )
print( res )
```

```
2 3
0 20
[[10  7  5]
 [13 17 10]]
```

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

In [89]:
```
import numpy as np

def generate_linspace(start, stop, num):
    linspace_array = np.linspace(start, stop, num)
    return linspace_array

start, stop, num = map( int, input().split() )
linspace_array = generate_linspace(start, stop, num)
print("Array of evenly spaced numbers over the specified range:", linsp
```

```
0 10 5
Array of evenly spaced numbers over the specified range: [ 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

In [90]:
```
import numpy as np

start, stop, num = 1, 100, 10
linspace_array = np.linspace( start, stop, num )
print( linspace_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

In [91]:
```
1 import numpy as np
2
3 start, stop, step = 2, 21, 2
4 res = np.arange( start, stop, step )
5 print( res )
```

```
[ 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

In [92]:
```
1 import numpy as np
2
3 start, stop, step = 1, 11, 0.5
4 res = np.arange( start, stop, step )
5 print( res )
```

```
[ 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]
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```