Exercise 1: Create a function in Python

Write a program to create a function that takes two arguments, name and age, and print their value.

Hint

- Use the `def` keyword with the function name to define a function.
- Next, take two parameters
- Print them using the print() function
- Call function by passing name and age.

Exercise 2: Create a function with variable length of arguments

Write a program to create function `func1()` to accept a variable length of arguments and print their value.

Note: Create a function in such a way that we can pass any number of arguments to this function, and the function should process them and display each argument's value.

Read: variable length of arguments in functions

Function call:

```
# call function with 3 arguments
func1(20, 40, 60)

# call function with 2 arguments
func1(80, 100)
```

Expected Output:

```
Printing values
20
40
60


Printing values
80
100
```

Hint

To accept a variable length of positional arguments, i.e., To create functions that take n number of positional arguments we use `*args` as a parameter. (prefix a parameter name with an asterisk *).

Using this, we can pass any number of arguments to this function. Internally all these values are represented in the form of a tuple.

Exercise 3: Return multiple values from a function

Write a program to create function `calculation()` such that it can accept two variables and calculate addition and subtraction. Also, it must **return both addition and subtraction in a single return call**.

Given:

```
def calculation(a, b):
    # Your Code

res = calculation(40, 10)
print(res)
```

Expected Output:

```
50, 30
```

Hint
Separate return values with a comma.

Exercise 4: Create a function with a default argument

Write a program to create a function `show_employee()` using the following conditions.

- It should accept the employee's name and salary and display both.
- If the salary is missing in the function call then assign default value 9000 to salary

Given:

```
showEmployee("Ben", 12000)
showEmployee("Jessa")
```

Expected output:

```
Name: Ben salary: 12000
Name: Jessa salary: 9000
```

Exercise 5: Create an inner function to calculate the addition in the following way

- Create an outer function that will accept two parameters, `a` and `b`
- Create an inner function inside an outer function that will calculate the addition of `a` and `b`
- At last, an outer function will add 5 into addition and return it

Exercise 6: Create a recursive function

Write a program to create a **recursive function to calculate the sum of numbers** from 0 to 10.

A recursive function is a function that calls itself again and again.

Expected Output:

55

Exercise 7: Assign a different name to function and call it through the new name

Below is the function `display_student(name, age)`. Assign a new name `show_tudent(name, age)` to it and call it using the new name.

Given:

```
def display_student(name, age):
    print(name, age)

display_student("Emma", 26)
```

Hint
Assign a different name to function using the assignment (=) operator.

`fun_name = new_name`

Exercise 8: Generate a Python list of all the even numbers between 4 to 30

Expected Output:

```
[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
```

Hint

- Use the built-in function range() to generate the sequence of numbers between the given start number to the stop number with a `step = 2` to get even numbers.
- pass `range()` function to a list constructor to create a list

Exercise 9: Find the largest item from a given list

```
x = [4, 6, 8, 24, 12, 2]
```

Expected Output:

```
24
```

Hint

Use the built-in function `max()` to get the largest number from a list