


Day2 programs to execute in Mojo

1. WAP to find the factorial of a given using the recursion function.


MOJO > DAY-2 >  facr_rec.mojo

```
1  fn fact(n: Int) -> Int:
2      if n <=1:
3          return 1
4      else:
5          return n*fact(n-1)
6
7  fn main():
8      var number: Int = 5
9      var result: Int = fact(number)
10     print(result)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo facr_rec.mojo
120
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

2. Define a global variable and show how it can be accessed and modified from different functions. Also, define a local variable within a function and demonstrate how it retains its value between function calls but is not accessible outside the function.

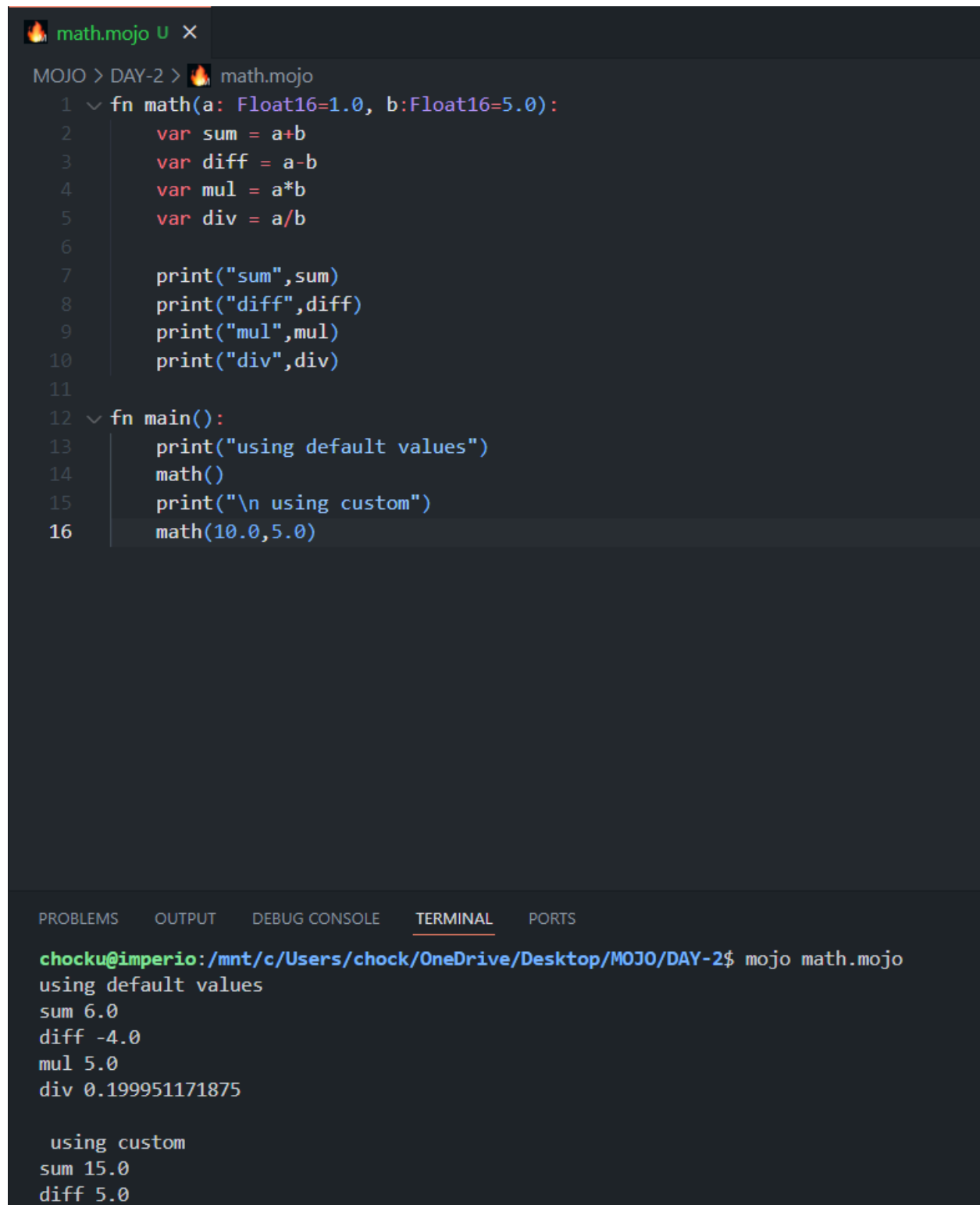
```
MOJO > DAY-2 >  glob.mojo
```

```
1  var glob = 20
2
3  def func():
4      a = 1
5      print("a inside func")
6
7  def main():
8      print("accesing global x",glob)
9      a = 10
10     func()
11     a+=a
12     func()
13     print("a outside function",a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo glob.mojo
accesing global x 20
a inside func
a inside func
a outside function 20
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

3. Write an example program for declaring arguments with a default value and perform some mathematical operations.



The screenshot shows a code editor with a file named `math.mojo`. The code defines a function `math` with default arguments `a: Float16=1.0` and `b: Float16=5.0`. It calculates `sum`, `diff`, `mul`, and `div`. A `main` function calls `math` with default values and then with custom values `10.0` and `5.0`.


```
1  fn math(a: Float16=1.0, b:Float16=5.0):
2      var sum = a+b
3      var diff = a-b
4      var mul = a*b
5      var div = a/b
6
7      print("sum",sum)
8      print("diff",diff)
9      print("mul",mul)
10     print("div",div)
11
12  fn main():
13     print("using default values")
14     math()
15     print("\n using custom")
16     math(10.0,5.0)
```

The terminal output shows the execution of the program. It first prints "using default values" followed by the results of the `math` function with default arguments. Then it prints "\n using custom" followed by the results of the `math` function with custom arguments.

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo math.mojo
using default values
sum 6.0
diff -4.0
mul 5.0
div 0.199951171875

using custom
sum 15.0
diff 5.0
```

4. WAP to add two numbers using normal parameter declaration, inout, and borrowed argument convention. Also, print the original values in the main code after calling each function.


MOJO > DAY-2 >  ex.mojo

```
1  fn add(x:Int, y:Int)->Int:
2      return(x+y)
3
4  fn borrow(borrowed x: Int, borrowed y: Int)->Int:
5      return x+y
6
7  fn add_inout(inout x:Int, inout y:Int)->Int:
8      x=20
9      y=10
10     return x+y
11
12  fn main():
13     var x:Int = 2
14     var y:Int = 4
15     print("after calling func",x,y)
16     print(borrow(x,y))
17     print("after calling fun",x,y)
18     print(add_inout(x,y))
19     print("after calling func",x,y)
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo ex.mojo
after calling func 2 4
6
after calling fun 2 4
30
after calling func 20 10
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

5. WAP to demonstrate struct to a print student details with a constructor.


MOJO > DAY-2 >  struct.mojo

```
1  struct Student:
2      var name:String
3      var age:Int
4      var grade:String
5      def __init__(inout self, name:String, age:Int,grade:String):
6          self.name = name
7          self.age = age
8          self.grade = grade
9      def print_deets(self):
10         print("Student deets:")
11         print("name",self.name)
12         print("age",self.age)
13         print("grade",self.grade)
14
15  def main():
16      var stu1: Student = Student(name="Alice",age=15,grade="A")
17      var stu2: Student = Student(name="Alex",age=20,grade="A")
18
19      stu1.print_deets()
20      stu2.print_deets()
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo struct.mojo
Student deets:
name Alice
age 15
grade A
Student deets:
name Alex
age 20
grade A
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

6. WAP to demonstrate a constructor for allocating the memory and a special function to deallocate the memory using struct.


MOJO > DAY-2 >  const.mojo

```
1  @value
2  struct Friend:
3      var name:String
4      var age: Int
5
6      fn __init__(inout self, name: String, age: Int):
7          self.name = name
8          self.age = age
9      fn __del__(owned self):
10         print("deleted self",self.name)
11
12  fn friends():
13      var fri = Friend("max",20)
14      print(fri.name)
15      print(fri.age)
16
17  def main():
18      friends()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio: /mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo const.mojo
max
20
deleted self max
chocku@imperio: /mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

7. WAP to demonstrate multiple constructors with the help of method overloading.

MOJO > DAY-2 >  mulconst.mojo

```
1  struct Student:
2      var name:String
3      var sem: Int
4
5      fn __init__(inout self):
6          self.name = "Name not found"
7          self.sem = 0
8
9      fn __init__(inout self, name:String, sem:Int):
10         self = Student()
11         self.name = name
12         self.sem = sem
13
14 def main():
15
16     var stud1 = Student()
17     print(stud1.name)
18     print(stud1.sem)
19
20     var stud2 = Student("Chocku",6)
21     print(stud2.name)
22     print(stud2.sem)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo mulconst.mojo
Name not found
0
Chocku
6
chocku@imperio:/mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```

8. Write example program to demonstrate `__copyinit__()` constructor.

```
MOJO > DAY-2 > 🔥 copy.mojo
1  struct Book:
2      var title: String
3      var author: String
4
5      fn __init__(inout self, title: String, author: String):
6          self.title = title
7          self.author = author
8      fn __copyinit__(inout self, existing: Self):
9          self.title = existing.title
10         self.author = existing.author
11
12 def main():
13     var originalBook = Book("Harry Potter", "JK Rowling")
14
15     var copied = originalBook
16
17     print(copied.title)
18     print(originalBook.title)
19
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
chocku@imperio: /mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$ mojo copy.mojo
Harry Potter
Harry Potter
chocku@imperio: /mnt/c/Users/chock/OneDrive/Desktop/MOJO/DAY-2$
```