1.

```
#q1
fn factorial(n:Int)->Int:
    if (n==1):
        return 1
    else:
        return n*factorial(n-1)
print(factorial(5))
```

[4] ✓ 0.2s mojo

```
120
```

2.

3.

```
#Q3
fn mathoperation(x:Int=5,y:Int =10):
    print("addition",x+y)
    print("multiplication",x*y)
mathoperation()
print("new call")
mathoperation(10,10)
```

[11] ✓ 0.3s mojo

```
addition 15
multiplication 50
new call
addition 20
multiplication 100
```

4.

```
#Q4
def addnuminout(inout x:Int,inout y:Int):
    x=1;y=2
    return x+y
a=10;b=20
print("Initial value of variables",a,b)
print(addnuminout(a,b))
print("Variables in main now",a,b)

def addnumborrow(borrowed x:Int,borrowed y:Int):
    x=5;y=6
    print("value in borrow function",x,y)
    return x+y
print(addnumborrow(a,b))
print("value outside borrow function",a,b)
```

✓ 1.3s mojo

```
Initial value of variables 10 20
3
Variables in main now 1 2
value in borrow function 5 6
11
value outside borrow function 1 2
```

5.

```
#Q5
struct student:
    var name:String
    var section:String
    def __init__(inout self, name:String,section:String):
        self.name=name
        self.section=section
    def printdetails(self):
        print("Student details are:")
        print(self.name,self.section)

var stud=student("richard","E")
stud.printdetails()
```

[21] ✓ 0.3s mojo

```
Student details are:
richard E
```

6.

```
#6
@value
struct MyPet:
    var name: String
    var age: Int

    fn __init__(inout self, name: String, age: Int):
        self.name = name
        self.age = age
    fn __del__(owned self):
        print("deleted self", self.name)


fn pets():
    var pet2 = MyPet("cat", 4)
    print(pet2.name)
    print(pet2.age)

pets()
```

[40] ✓ 0.5s

```
cat
4
deleted self cat
```

7.

```
#7
struct MyPet:
    var name: String
    var age: Int

    fn __init__(inout self):
        self.name = "name not found"
        self.age = 0

    fn __init__(inout self, name: String, age: Int):
        self = MyPet()
        self.name = name
        self.age = age

var pet = MyPet()
var pet2 = MyPet("cat", 4)

print(pet.name)
print(pet.age)

print(pet2.name)
print(pet2.age)
```

[26] ✓ 0.5s

```
name not found
0
cat
4
```

8.

```
#10
struct Book:
    var title: String
    var author: String
    fn __init__(inout self, title: String, author: String):
        self.title = title
        self.author = author

    fn __copyinit__(inout self, existing: Self):
        self.title = existing.title
        self.author = existing.author

var originalBook = Book("Feluda", "Satyajit Ray")

# using copy constructor
var copiedBook = originalBook

print(copiedBook.title)
```

[15] ✓ 0.4s

```
Feluda
```