

Lab 9.2 (Deadline Monday 21 March 23:59)

- Upload your code to [Einstein](#) to have it verified.

Sum up

- In a module named `sumup_092.py` implement a `sumup()` function that when passed an integer `n`, returns the sum of all numbers 0 through `N`.
- **Your function must be recursive.**
- When implemented correctly, running the following program should produce the given output.

```
from sumup_092 import sumup

def main():
    print(sumup(0))
    print(sumup(1))
    print(sumup(12))

if __name__ == '__main__':
    main()
```

```
0
1
78
```

Factorial

- In a module named `factorial_092.py` implement a `factorial()` function that when passed an integer `n`, returns $n!$ (i.e. $n * n-1 * n-2 * \dots * 2 * 1$).
- **Your function must be recursive.**
- When implemented correctly, running the following program should produce the given output.

```
from factorial_092 import factorial

def main():
    print(factorial(0))
    print(factorial(1))
    print(factorial(12))

if __name__ == '__main__':
    main()
```

```
1
1
479001600
```

Power

- In a module named `power_092.py` implement a `power()` function that when passed two integers `m`, and `n`, returns `m` to the power of `n`.
- **Your function must be recursive.**
- When implemented correctly, running the following program should produce the given output.

```
from power_092 import power

def main():
    print(power(2,3))
    print(power(4,4))
    print(power(2,32))
    print(power(10,3))
    print(power(8,0))

if __name__ == '__main__':
    main()
```

```
8
256
4294967296
1000
1
```

Minimum

- In a module named `minimum_092.py` implement a `minimum()` function that when passed a list of integers returns the minimum integer in the list.
- **Your function must be recursive and cannot use the built-in `min()` function.**
- When implemented correctly, running the following program should produce the given output.

```
from minimum_092 import minimum

def main():
    min = None
    print(minimum([6,5,1,3,4]))
    print(minimum([6,5,11,3,4]))
    print(minimum([6,15,11,13,14]))
    print(minimum([6,15,11,13,4]))

if __name__ == '__main__':
    main()
```

```
1
3
6
4
```

Maximum

- In a module named `maximum_092.py` implement a `maximum()` function that when passed a list of integers returns the maximum integer in the list.
- **Your function must be recursive and cannot use the built-in `max()` function.**
- When implemented correctly, running the following program should produce the given output.

```
from maximum_092 import maximum

def main():
    max = None
    print(maximum([6,5,1,3,4]))
    print(maximum([6,5,11,3,4]))
    print(maximum([6,15,11,13,14]))
    print(maximum([6,15,11,13,4]))

if __name__ == '__main__':
    main()
```

```
6
11
15
15
```

Count

- In a module named `count_092.py` implement a `count_letters()` function that when passed a string returns the number of characters in the string.
- **Your function must be recursive and cannot use the built-in `len()` function.**
- When implemented correctly, running the following program should produce the given output.

```
from count_092 import count_letters

def main():
    len = None
    print(count_letters('cat'))
    print(count_letters('catastrophe'))
    print(count_letters(''))

if __name__ == '__main__':
    main()
```

```
3
11
0
```

Reversing a list

- In a module named `reverse_092.py` implement a `reverse_list()` function that returns a new list that is the reverse of the list passed to it.
- **Your function must be recursive.**
- When implemented correctly, running the following program should produce the given output.

```
from reverse_092 import reverse_list

def main():
    print(reverse_list([1,2,3]))
    print(reverse_list([3,4,5,6]))
    print(reverse_list([1,2]))

if __name__ == '__main__':
    main()
```

```
[3, 2, 1]
[6, 5, 4, 3]
[2, 1]
```

Fibonacci

- In a module named `fibonacci_092.py` implement a `fibonacci()` function that returns the *Nth* number in the Fibonacci sequence.
- **Your function must be recursive.**
- When implemented correctly, running the following program should produce the given output.

```
from fibonacci_092 import fibonacci

def main():
    print(fibonacci(0))
    print(fibonacci(1))
    print(fibonacci(5))
    print(fibonacci(8))

if __name__ == '__main__':
    main()
```

```
1
1
8
34
```