```
#!/usr/bin/python3
from pylab import *
from math import *
from numpy import *

def p1():
    """Fibonacci Sequence: Prints all the sequence from 1 to Max"""
    print("Fibonacci Sequence:")
    vmaximum = int(input(’Max: ’))
    fibs = [1, 1]
    while fibs[-1] < vmaximum:
        fibs.append(fibs[-1] + fibs[-2])
    if fibs[-1] > vmaximum:
        fibs.pop()
    print(*fibs)

if __name__ == "__main__":
    p1()
```

```python
#!/usr/bin/python3
from pylab import *
from math import *
from numpy import *

def p2():
    """2.9: The Madelung Constant"""
    print("2.9 [The madelung constant]:")
    limit = int(input("L: "))

    def v(i, j, k):
        """Calculates the \"Potential\" created by an atom"""
        return (-1 if
                (i + j + k) % 2 == 1 else 1) * 1 / sqrt(i**2 + j**2 + k**2)

    madelung = 0
    for i in range(-limit, limit + 1):
        for j in range(-limit, limit + 1):
            for k in range(-limit, limit + 1):
                if i == j == k == 0:
                    continue
                madelung += v(i, j, k)
    print(madelung)

if __name__ == "__main__":
    p2()
```

```
#!/usr/bin/python3
from pylab import *
from math import *
from numpy import *

def p3():
    """2.10: the Semi-Emperical Mass Formula"""

    print("2.10 [The semi-emperical mass formula]:")
    def calc_b(A, Z):
        return (15.8 * A) - (18.3 * pow(A, 2 / 3)) - (0.714 * (
            pow(Z, 2) / pow(A, 1 / 3))) - (23.2 * (pow(A - 2 * Z, 2) / A)) + (
                (0 if A % 2 == 1 else
                 (12.0 if Z % 2 == 0 else -12.0)) / pow(A, 1 / 2))

    def a():
        print("2.10.a)")
        a = int(input("A: "))
        z = int(input("Z: "))
        print(calc_b(a, z))

    def b():
        print("2.10.b)")
        a = int(input("A: "))
        z = int(input("Z: "))
        print(calc_b(a, z) / a)

    def c():
        print("2.10.c)")
        z = int(input("Z: "))
        bn = []
        for a in range(z, 3 * z + 1):
            bn.append(calc_b(a,z) / a)
        print(z + bn.index(max(bn)), max(bn))

    def d():
        print("2.10.d)")
        zbn = []
        for z in range(1, 100):
            bn = []
            for a in range(z, 3 * z + 1):
                bn.append(calc_b(a,z)/a)
            print("{}:".format(z), z + bn.index(max(bn)), max(bn))
            zbn.append(max(bn))
        print("Max at:", 1 + zbn.index(max(zbn)), max(zbn))
    a()
    b()
    c()
    d()

if __name__ == "__main__":
    p3()
```

```
#!/usr/bin/python3
from pylab import *
from math import *
from numpy import *

def p4():
    """2.12: Prime Numbers"""
    print("2.12 [Prime Numbers]:")
    max_prime = int(input("Max: "))
    primes = [2]

    def is_prime(n):
        sqrt_n = sqrt(n)
        for p in primes:
            if p > sqrt_n:
                return True
            if n % p == 0:
                return False
    for n in range(3, max_prime, 2):
        if is_prime(n):
            primes.append(n)
    print(", ".join([str(x) for x in primes]))

if __name__ == "__main__":
    p4()
```

```
#!/usr/bin/python3
from pylab import *
from math import *
from numpy import *

def p5():
    """3.3"""
    print("3.3:")
    data = loadtxt("stm.txt")
    imshow(data, origin="lower")
    set_cmap('Greens_r')
    show()
    imshow(data, origin="lower")
    set_cmap('flag')
    show()
    maps = colormaps()
    # for i in maps:
    #     imshow(data, origin="lower")
    #     xlabel(i)
    #     set_cmap(i)
    #     show()

if __name__ == "__main__":
    p5()
```