

Name: _____

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing.

1. The function `wrmat()` inputs a matrix and a file name, and outputs the matrix to a text file. For example, the call `wrmat([[1,2,3], [5,12,13]], 'outmat.txt')` would produce the text file `outmat.txt` with contents

```
1 2 3
5 12 13
```

Fill in the blanks.

```
def wrmat(mat, tf):
    f = open(tf, 'w')
    for row in mat:
        # insert 1 or more lines
        f.write(outrow+'\n')
    f.close
```

2. The function `primefact()` below finds the prime factorization of number, relative to the given primes. For example, the call `primefact([2,3,5,7,11],24)` would return `[2,3], [3,1]`, meaning that $24 = 2^3 3^1$. (It is assumed that the prime factorization of `n` does indeed exist for the numbers in `primes`.) Fill in the blanks.

```
# find the maximal power of p
# that evenly divides m
def dividetomax(p,m):
    k = 0
    while True:
        if m % p != 0: # blank
            k += 1
            m /= p
```

```
def primefact(primes, n):
    tmp = map( # blank
    tmp = filter( # blank
    return tmp
```

Solutions:

1.

```
def wrmat(mat, tf):
    f = open(tf, 'w')
    for row in mat:
        outrow = ''
        for elt in row:
            outrow += str(elt) + ' '
        f.write(outrow+'\n')
    f.close
```

2.

```
# find the maximal power of p
# that evenly divides m
def dividetomax(p,m):
```

```
k = 0
while True:
    if m % p != 0: return (p,k)
    k += 1
    m /= p

def primefact(primes, n):
    tmp = map(dividetomax, primes, len(primes)*[n])
    tmp = filter(lambda u: u[1] > 0, tmp)
    return tmp
```