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. (10) Below is a generator version of the circular queue example on p.85, plus a test program. Fill in the blanks:

```python
def cq(q):
    while True:
        # one blank line
        # one blank line

def main():
    x = [5,12,13]
    g = cq(x)
    print g.next()  # prints 5
    print g.next()  # prints 12
    print g.next()  # prints 13
    print g.next()  # prints 5
    print g.next()  # prints 12
```

2. (10) Below is a function to find all subsets of size $k$ from a set of size $n$. Here's a test:

```python
def subsets(n,k):
    # remaining code...

def main():
    n = int(sys.argv[1])
    k = int(sys.argv[2])
    g = subsets(n,k)
    for sub in g: print sub

% python subsets.py 5 2
[0, 1]
[0, 2]
[0, 3]
[0, 4]
[1, 2]
[1, 3]
[1, 4]
[2, 3]
[2, 4]
[3, 4]
```

Fill in the blanks:

```python
def subsets(n,k):
    if k == 0:
        yield # blank
        # blank
    for i in range(n-k+1):
        # find all subsets beginning with i
        g = # blank
        for sub in g:
            yield #blank
```
Solutions:

1.
```python
def cq(q):
    while True:
        head = q[0]
        yield head
        q = q[1:] + [head]

def main():
    x = [5, 12, 13]
    g = cq(x)
    print(g.next())
    print(g.next())
    print(g.next())
    print(g.next())
    print(g.next())

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

2.
```python
import sys

def subsets(n, k):
    if k == 0:
        yield []
        raise StopIteration
    for i in range(n-k+1):
        # find all the subsets beginning with i
        g = subsets(n-i-1, k-1)
        for sub in g:
            yield [i] + map(lambda u:u+i+1, sub)

def main():
    n = int(sys.argv[1])
    k = int(sys.argv[2])
    g = subsets(n, k)
    for sub in g:
        print(sub)

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