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:
        head = q[0]
        # 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.
1 def cq(q):
  2 while True:
  3     head = q[0]
  4     yield head
  5     q = q[1:] +[head]
  6  
  7 def main():
  8     x = [5,12,13]
  9     g = cq(x)
 10     print g.next()
 11     print g.next()
 12     print g.next()
 13     print g.next()
 14     print g.next()
 15
 16 if __name__ == '__main__': main()

2.
1 import sys
2 def subsets(n,k):
  3 if k == 0:
  4     yield []
  5     raise StopIteration
  6 for i in range(n-k+1):
  7     # find all the subsets beginning with i
  8     g = subsets(n-i-1,k-1)
  9     for sub in g:
 10         yield [i] + map(lambda u:u+i+1,sub)
 11
 12 def main():
 13     n = int(sys.argv[1])
 14     k = int(sys.argv[2])
 15     g = subsets(n,k)
 16     for sub in g: print sub
 17
 18 if __name__ == '__main__': main()