```
from collections import deque
from threading import Thread

class asynchronous(object):
  def __init__(self, func):
    self.func = func

    def threaded(*args, **kwargs):
        self.queue.append(self.func(*args, **kwargs))

    self.threaded = threaded

  def __call__(self, *args, **kwargs):
      return self.func(*args, **kwargs)

  def start(self, *args, **kwargs):
      self.queue = deque()
      thread = Thread(target=self.threaded, args=args, kwargs=kwargs);
      thread.start();
      return asynchronous.Result(self.queue, thread)

  class NotYetDoneException(Exception):
    def __init__(self, message): self.message = message

  class Result(object):
    def __init__(self, queue, thread):
      self.queue = queue
      self.thread = thread

    def is_done(self):
      return not self.thread.is_alive()

    def get_result(self):
      if not self.is_done():
          raise asynchronous.NotYetDoneException('the call has not yet complete
      if not hasattr(self, 'result'):
          self.result = self.queue.popleft()
      return self.result
```