```
#!/usr/bin/python

"""
Output lines selected randomly from a file
Copyright 2005, 2007 Paul Eggert.
Copyright 2010 Darrell Benjamin Carbajal.
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
Please see <http://www.gnu.org/licenses/> for a copy of the license.
$Id: randline.py,v 1.4 2010/04/05 20:04:43 eggert Exp $
"""

import random, sys
from optparse import OptionParser

class randline:
    def __init__(self, filename):
        f = open(filename, 'r')
        self.lines = f.readlines()
        f.close()

    def chooseline(self):
        return random.choice(self.lines)

def main():
    version_msg = "%prog 2.0"
    usage_msg = """%prog [OPTION]... FILE
Output randomly selected lines from FILE."""

    parser = OptionParser(version=version_msg,
                          usage=usage_msg)
    parser.add_option("-n", "--numlines",
                      action="store", dest="numlines", default=1,
                      help="output NUMLINES lines (default 1)")
    options, args = parser.parse_args(sys.argv[1:])

    try:
        numlines = int(options.numlines)
    except:
        parser.error("invalid NUMLINES: {0}".
                     format(options.numlines))
    if numlines < 0:
        parser.error("negative count: {0}".
                     format(numlines))
    if len(args) != 1:
        parser.error("wrong number of operands")
    input_file = args[0]

    try:
        generator = randline(input_file)
        for index in range(numlines):
            sys.stdout.write(generator.chooseline())
    except IOError as (errno, strerror):
        parser.error("I/O error({0}): {1}".
                     format(errno, strerror))

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