# Programming

## From 1 to 4

```
# You didn’t make this code
# IRC is annoying to use
# this solves all first 3 challenges just change run

# python <file> <challenge number>

#!/usr/local/bin/python
#cSpell:disable

import sys, socket, ssl
from time import sleep
from math import sqrt
import base64, codecs, zlib

server = "irc.root-me.org"
port = 6697

nick = "TestUser42"
password = "MYSECRETPASSWORD1337"

channel = "#root-me_challenge"
sendtobot = "candy"

timeout = 1 #seconds
retries = 5

DEBUG = False

challenge = 0 #dummy
try:
    challenge = 4 if len(sys.argv) == 1 else int(sys.argv[1])
    if challenge < 1 or challenge > 4:
        raise Exception('BadNum')
except Exception, e:
    raise RuntimeError('First and only argument must be integer! (challenge number, from 1 to 4)')

print('You selected {} challenge\n'.format(challenge))

class irc(object):
    socket = None
    srv = None
    port = None
    nick = None
    pwd = None
    channel = None
    isAuthorized = False

    def _check_auth(func):
        def wrapper(self, *arg, **kw):
```

```
        if self.isAuthorized:
            return func(self, *arg, **kw)
        else:
            raise Exception('Not authorized!')
        return None
    return wrapper

def __init__(self, srv, port):
    try:
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.socket = ssl.wrap_socket(self.socket)
        self.socket.settimeout(timeout)
        self.socket.connect((srv, port))
    except Exception, e:
        raise RuntimeError('Failed to connect to IRC server!')
    self.srv = srv
    self.port = port

def __del__(self):
    if self.socket:
        self._send('QUIT :bye bye!')
        self.socket.close
        self.socket = None

def _send(self, line, EOL = '\r\n'):
    if DEBUG:
        print('SEND:\n\t%s' % (line))
    self.socket.send(line + EOL)

def _recv(self, size = 8192):
    data = ''
    while True:
        try:
            tempdata = self.socket.recv(size)
        except ssl.SSLError:
            tempdata = None
        if not tempdata:
            break
        data += tempdata
    if DEBUG and data:
        print('RECV:\n\t%s' % (data))
    return data if data else None

def auth(self, nick, pwd, _check_auth_func, realname = None):
    self._send('NICK ' + nick)
    sleep(1)
    self._send('PASS ' + password)
    sleep(1)
    self._send('USER {0} {0} {0} :{1}'.format(nick, realname if realname else nick + '-Test'))
    sleep(1)
    self._send('PRIVMSG nickserv :identify {} {}'.format(nick, pwd))
    sleep(1)

    if not _check_auth_func(self._recv()):
        raise Exception('Auth Failed!')
    else:
        self.isAuthorized = True
        self.nick = nick
        self.pwd = pwd

    @_check_auth
    def join(self, channel):
        self._send('JOIN ' + channel)
        self.channel = channel

    @_check_auth
    def privmsg(self, reciever, msg):
        self._send("PRIVMSG {} :{}".format(reciever, msg))
        return self._recv()

    @_check_auth
    def ping_pong(self):
        reply = self._recv()
        if reply:
            if reply.find('PING') != -1:
                self._send('PONG ' + reply.split()[1])


print "Establishing connection to {}:{}...".format(server, port)
IRC = irc(server, port)
print "Connection established!\n"
print "Try to auth as {}...".format(nick)
IRC.auth(nick, password, lambda reply: True if reply.find(' :your unique ID') != -1 else False)
print "Auth succeed!\n"
print "Join channel {}".format(channel)
IRC.join(channel)
print "Start chat with bot '{}'\n".format(sendtobot)

def ep1(bot_reply):
    nums = [int(num) for num in bot_reply.split(' / ')]
    return round(sqrt(nums[0]) * nums[1], 2)

def ep2(bot_reply):
    return base64.b64decode(bot_reply)

def ep3(bot_reply):
    return codecs.decode(bot_reply, 'rot_13')

def ep4(bot_reply):
    return zlib.decompress(base64.b64decode(bot_reply))

retries_actual = retries
while retries_actual > 0:
    IRC.ping_pong()

    print('Attempt #{} (of {})'.format(retries - retries_actual + 1, retries))

    reply = IRC.privmsg(sendtobot, '!ep' + str(challenge)).split(':')[2]
    print('Challenge - {}'.format(reply.replace('\r\n', '')))

    ans = locals()['ep{}'.format(challenge)](reply)
    print('Answer - {}'.format(ans))
    reply = IRC.privmsg(sendtobot, '!ep{} -rep {}'.format(challenge, ans))

    if reply.find('You dit it!') != -1:
        print('SUCCESS!\n{}'.format(reply.split(':')[2]))
        break
    else:
        print('Failed!\n{}'.format(reply.split(':')[2]))
        if reply.find('BANNED') != -1:
            print 'Sleep for 30 seconds...'
            sleep(30)
        retries_actual -= 1

print('\n{0}\nHave a good day, h4x0r!\n{0}\n'.format('=' * 40))
```