0x0000000000001149 <+0>: endbr64

0x000000000000114d <+4>: push %rbp

0x000000000000114e <+5>: mov %rsp,%rbp

0x0000000000001151 <+8>: sub $0x20,%rsp

0x0000000000001155 <+12>: mov %edi,-0x14(%rbp)

0x0000000000001158 <+15>: mov %rsi,-0x20(%rbp)

**Setting things up ready ^**

0x000000000000115c <+19>: movl $0x0,-0xc(%rbp)

**Assigning value of x ^ i.e x=0xc**

0x0000000000001163 <+26>: movl $0x1,-0x8(%rbp)

**Assigning value of y ^ i.e y=0x8**

0x000000000000116a <+33>: mov -0xc(%rbp),%eax

0x000000000000116d <+36>: mov %eax,%esi

0x000000000000116f <+38>: lea 0xe8e(%rip),%rdi # 0x2004

0x0000000000001176 <+45>: mov $0x0,%eax

0x000000000000117b <+50>: callq 0x1050 <printf@plt>

**Print statement instructions ^ In callq it calling some memory address not mentioned here where value of x might be stored.**

0x0000000000001180 <+55>: mov -0xc(%rbp),%edx

**Moving the value of x to a register %edx**

0x0000000000001183 <+58>: mov -0x8(%rbp),%eax

**Moving the value of x to a register %eax**

0x0000000000001186 <+61>: add %edx,%eax

**Adding value of edx and eax and the added vaue sits in edx**

0x0000000000001188 <+63>: mov %eax,-0x4(%rbp)

**Then we move the value of %eax i.e 0x8 i.e y to a new register 0x4 which is probably the z**

0x000000000000118b <+66>: mov -0x8(%rbp),%eax

**Now we assign the value of eax to the value of 0x8 i.e y**

0x000000000000118e <+69>: mov %eax,-0xc(%rbp)

**After that we assign the value of eax which actually contains the value of y to 0xc i.e x**

0x0000000000001191 <+72>: mov -0x4(%rbp),%eax

**Then we assign the value of 0x4 i.e z to eax**

0x0000000000001194 <+75>: mov %eax,-0x8(%rbp)

**Then we assign the value of eax i.e z to 0x8 i.e y**

0x0000000000001197 <+78>: cmpl $0xfe,-0xc(%rbp)

**After that compare between 0xfe i.e 255 with 0xc i.e x**

0x000000000000119e <+85>: jle 0x116a <main+33>

**If second is less than first (which is implemented by the command jle [jump less than]) then jump to line 0x116a**

0x00000000000011a0 <+87>: jmp 0x115c <main+19>+

**Else normally jump to line 0x115c**

**So overall we got that ->**

**0xc = x**

**0x8 = y**

**0x4 = z**

**And some temporary registers like eax, edx**