

1

对于变量的赋值，汇编中使用 `movb`、`movw` 和 `movl` 对应于 1 字节、2 字节和 4 字节长度变量的赋值，使用 `mov* $*, *(%rbp)` 语法。

对于指针的赋值，汇编中先使用 `leaq *(%rap), %rax` 将变量的地址赋给寄存器 `rax`，然后使用 `movq %rax, *(%rbp)` 指令将 8 字节长度的地址从寄存器 `rax` 中取出并赋给相应的其他寄存器。

最后会有指令执行 `return 0` 的语句。

2

对于一个数组使用常量下标赋值，与正常变量赋值一致，使用 `mov*` 指令，对于数组中变量使用常量下标打印，会将数据存放到 `eax` 寄存器和 `edx` 寄存器，然后使用 `leaq .LC0(%rip), %rcx` 指令赋值，最后 `call printf` 执行 `printf` 函数。

对于变量下标，会在 `mov*` 指令中加入变量值作为地址偏移量，其他与常量下标一致。

3

使用指针访问时，常量指针下标与常量下标一致，而使用变量指针下标时，会使用两次 `leaq` 指令和一次 `addq` 指令计算地址偏移量，并将地址存到寄存器中，然后使用 `mov` 指令将值赋到寄存器中。在进行打印时，也会先计算地址偏移量，然后再取出，进入 `printf` 指令。