

Exercise Sheet

10/10

1. 请描述下面程序的表达式拆分结果。

```
n = 10;
i = 0;
while (i < n && * (p + i * 8) != 0) do {
    i = i + 1
}
```

2. 请描述下面 while 语言程序对应的控制流图，即每个基本块中的指令与跳转信息。

```
x = 10;
while (x > 0) do {
    x = x - 1
};
y = x
```

3. 请对下面程序做 liveness 分析，对于每一条语句 u 写出 $def(u)$ 、 $use(u)$ 、 $in(u)$ 与 $out(u)$ 。

```
Block 0:
    #2 = read_int()
    #4 = 0
    jmp 1
Block 1:
    if (NE(#2, 0)) then jmp 2 else jmp 3
Block 2:
    #7 = Deref(#2)
    #4 = PLUS(#4, #7)
    #8 = PLUS(#2, 8)
    #2 = Deref(#8)
    jmp 1
```

4. 请对下面程序做 liveness 分析，对于每一条语句 u 写出 $def(u)$ 、 $use(u)$ 、 $in(u)$ 与 $out(u)$ 。

```
Block 0:
    #3 = 0
    #2 = read_int()
    jmp 1
Block 1:
    if (NE(#2, 0)) then jmp 2 else jmp 3
Block 2:
    #7 = #2
    #8 = PLUS(#2, 8)
    #2 = Deref(#8)
    #9 = PLUS(#7, 8)
    * #9 = #3
    #3 = #7
    jmp 1
```

5. 考虑下面基本块程序：

```
Block 0:
  #0 = read_int()
  #1 = 1
  #2 = 0
  #5 = read_int()
  #6 = read_int()
  #7 = read_int()
  #8 = read_int()
  jmp 1
Block 1:
  if (GT(#0, 0)) then jmp 2 else jmp 3
Block 2:
  #9 = MUL(#1, #5)
  #10 = MUL(#2, #7)
  #3 = PLUS(#9, #10)
  #11 = MUL(#1, #6)
  #12 = MUL(#2, #8)
  #4 = PLUS(#11, #12)
  #1 = #3
  #2 = #4
  #0 = MINUS(#0, 1)
  jmp 1
```

假设 `read_int()` 的返回值总是使用 `rax` 寄存器存储，寄存器数量 $K=9$ （包括 `rax` 寄存器在内，有 9 个可以用于寄存器分配的寄存器）。请根据上课所学寄存器分配算法，对这段程序中的 13 个变量进行寄存器分配。本题需要写出必要的中间结果，如：liveness 分析以及 simplify、spill、select 等步骤的选择。

6. 假设寄存器数量 $K=8$ （除了 `rbp`、`pc` 等不能表示变量数值的寄存器之外，包括 `rax` 寄存器在内，有 8 个寄存器），请根据上课所学寄存器分配算法，对这段程序中的变量进行寄存器分配。在无法 Simplify 时，假定总是选择 Spill 编号最小的剩余变量。本题需要写出必要的中间结果，如：liveness 分析以及 simplify、spill、select 等步骤的选择。