

◆ .gitignore Loop00.ll ① README.md

TEST > Exercise1 > Loop00.ll

```

9  define i32 @g_incr(i32 noundef %0) #0 {           ; int g_incr(int c)
10      ; %1 = BB1
11      %2 = alloca i32, align 4                     ; int %2
12      store i32 %0, ptr %2, align 4                ; %2 = c
13      %3 = load i32, ptr %2, align 4                ; %3 = %2
14      %4 = load i32, ptr @g, align 4                ; %4 = g
15      %5 = add nsw i32 %4, %3                      ; %5 = %4 + %3 -> %5 = g + c
16      store i32 %5, ptr @g, align 4                ; g = %5
17      %6 = load i32, ptr @g, align 4                ; %6 = g
18      ret i32 %6                                   ; return %6 -> return g
19  }
20
21  ; Function Attrs: noinline nounwind optnone ssp uwtable(sync)
22  define i32 @loop(i32 noundef %0, i32 noundef %1, i32 noundef %2) #0 { ; int loop(int a, int b, int c)
23      ; %3 = BB2
24      %4 = alloca i32, align 4                     ; int %4
25      %5 = alloca i32, align 4                     ; int %5
26      %6 = alloca i32, align 4                     ; int %6
27      %7 = alloca i32, align 4                     ; int %7 -> int i
28      %8 = alloca i32, align 4                     ; int %8
29      store i32 %0, ptr %4, align 4                 ; %4 = a
30      store i32 %1, ptr %5, align 4                 ; %5 = b
31      store i32 %2, ptr %6, align 4                 ; %6 = c
32      store i32 0, ptr %8, align 4                 ; %8 = 0 -> ret = 0
33      %9 = load i32, ptr %4, align 4                 ; %9 = %4 -> %9 = a
34      store i32 %9, ptr %7, align 4                 ; %7 = %9 -> i = a
35      br label %10                                ; unconditional jump to label 10
36
37  10:                                             ; preds = %17, %3
38      ; %10 = BB3
39      %11 = load i32, ptr %7, align 4                ; %11 = %7 -> %11 = i
40      %12 = load i32, ptr %5, align 4                ; %12 = %5 -> %12 = b
41      %13 = icmp slt i32 %11, %12                  ; %13 = %11 < %12 -> %13 = i < b
42      br i1 %13, label %14, label %20                ; jump based on %13 jump to label 14 or 20
43
44  14:                                             ; preds = %10
45      ; %14 = BB4
46      %15 = load i32, ptr %6, align 4                ; %15 = %6 -> %15 = c
47      %16 = call i32 @g_incr(i32 noundef %15)        ; %16 = g_incr(%15) -> %16 = g_incr(c)
48      br label %17                                ; unconditional jump to label 17 (fallthrough)
49
50  17:                                             ; preds = %14
51      ; %17 = BB5
52      %18 = load i32, ptr %7, align 4                ; %18 = %7 -> %18 = i
53      %19 = add nsw i32 %18, 1                      ; %19 = %18 + 1 -> %19 = i + 1
54      store i32 %19, ptr %7, align 4                ; %7 = %19 -> i = i + 1
55      br label %10, !llvm.loop !6                  ; unconditional jump to label 10
56
57  20:                                             ; preds = %10
58      ; %20 = BB6
59      %21 = load i32, ptr %8, align 4                ; %21 = %8 -> %21 = ret
60      %22 = load i32, ptr @g, align 4                ; %22 = g
61      %23 = add nsw i32 %21, %22                    ; %23 = %21 + %22 -> %23 = ret + g
62      ret i32 %23                                  ; return %23
63  }
64

```

BB1

BB2

FT

BB3

FT

T

BB4

FT

F

BB5

BB6