

循环结构

三种循环在Debug下结构很明显，而在Release下均会转换为等价的 do-while

且 continue 和 break 必定会有条件跳转，跳转到开头则是 continue，跳转到末尾则是 break

未优化

do-while

```
DO_BEGIN:
    ; 循环体
    ...
    ...
    jxx DO_BEGIN:    ; 上跳
```

while

```
WHILE_BEGIN:
    jxx WHILE_END    ; 下跳到while结尾
    ; 循环体
    ...
    ...
    jmp WHILE_BEGIN  ; 上跳到while开始
WHILE_END:
```

for

```
    mov mem/reg, xxx    ; for初始化部分，即表达式1
    jmp FOR_CMP
FOR_STEP:                ; for修改步长部分，即表达式3
    mov reg, STEP
    add reg, xxx
    mov STEP, reg

FOR_CMP:                ; for比较部分，即表达式2
    ; 循环体
    mov ecx, dword ptr STEP
    cmp ecx, STEPEND
    jxx FOR_END
    ...
    ...
    jmp FOR_STEP
FOR_END:
```

优化

1. 转化为等价 do-while

```

while(n <= 100) {
    sum = sum + n;
    n++;
}

// 等价do-while
if(n <= 100) {
    do {
        sum = sum + n;
        n++;
    } while(n <= 100)
}

```

2. 代码外提

- 判定条件如果是函数调用，则函数调用不会代码外提，在条件处插入内联函数实现

3. 强度削弱

```

while(argc <= 100) {
    sum = argc * n;
    argc++;
}

// 强度削弱
sum = argc * n;
edx = 100 - argc;
do {
    sum = sum + n;
    edx--;
} while(edx > 0);

```