What is a compiler?
A compiler is a program translator.

(high-level) source language code

Compiler

(low-level) target language code
int expr(int n) {
    int d = 4 * n * n * (n + 1) * (n + 1);
    return d;
}

lda $30,-32($30)
stq $26,0($30)
stq $15,8($30)
bis $30,$30,$15
bis $16,$16,$1
stl $1,16($15)
lds $f1,16($15)
sts $f1,24($15)
ldl $5,24($15)
bis $5,$5,$2
s4addq $2,0,$3
ldl $4,16($15)
mull $4,$3,$2
ldl $3,16($15)

addq $3,1,$4
mull $2,$4,$2
ldl $3,16($15)
addq $3,1,$4
mull $2,$4,$2
ldl $3,16($15)
addq $3,1,$4
mull $2,$4,$2
stl $2,20($15)
ldl $0,20($15)
br $31,$33
bis $15,$15,$30
ldq $26,0($30)
ldq $15,8($30)
addq $30,32,$30
ret $31,($26),1
A compiler is a program improver.

Unoptimized Code

```
lda $30,-32($30)
stq $26,0($30)
stq $15,8($30)
bis $30,$30,$15
bis $16,$16,$1
stl $1,16($15)
lds $f1,16($15)
sts $f1,24($15)
ldl $5,24($15)
bis $5,$5,$2
s4addq $2,0,$3
ldl $4,16($15)
mull $4,$0,$0
mull $0,$16,$0
mull $0,$16,$0
ret $31,($26),1
```

Optimized Code

```
s4addq $16,0,$0
mull $16,$0,$0
addq $16,1,$16
mull $0,$16,$0
mull $0,$16,$0
ret $31,($26),1
```

$33:

```
bis $15,$15,$30
ldq $26,0($30)
ldq $15,8($30)
addq $30,32,$30
ret $31,($26),1
```
Simplified Compiler Structure

Source code
if (b == 0) a = b;

Understand source code

Intermediate code

Optimize

Intermediate code

Generate assembly code

Assembly code
cmp $0,%ecx
cmovz %edx,%ecx

Front end
(machine-independent)

Optimizer

Back end
(machine-dependent)
Compiler Front End

Source code (character stream)

if (b == 0) a = b;

Token stream

Syntax Analysis (Parsing)

Abstract syntax tree (AST)

if (b == 0) a = b;

Decorated AST

boolean = if = int

int b int 0 int a int b

Semantic Analysis

Lexical Analysis
Compiler Middle / Back End

Decorated AST

Intermediate code

Assembly code

Intermediate Code Generation

Optimizations

Machine Optimizations and Code Generation

\[
t = (b == 0)
falsejump \ t, L
a = b
label \ L
\]

\[
t = (b == 0)
falsejump \ t, L
a = 0
label \ L
\]

cmp $0,ecx
cmovz edx,ecx