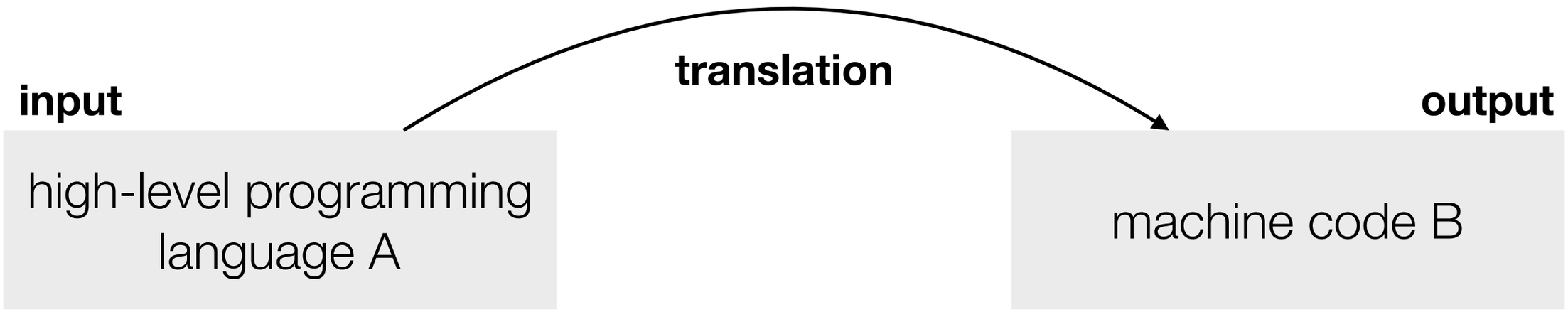
# The Selfie Library

```
uint64_t leftShift(uint64_t n, uint64_t b);
uint64_t rightShift(uint64_t n, uint64_t b);

uint64_t getBits(uint64_t n, uint64_t i, uint64_t b);
uint64_t getLowWord(uint64_t n);
uint64_t getHighWord(uint64_t n);

uint64_t abs(uint64_t n);

uint64_t signedLessThan(uint64_t a, uint64_t b);
uint64_t signedDivision(uint64_t a, uint64_t b);

uint64_t isSignedInteger(uint64_t n, uint64_t b);
uint64_t signExtend(uint64_t n, uint64_t b);
uint64_t signShrink(uint64_t n, uint64_t b);

uint64_t  loadCharacter(uint64_t* s, uint64_t i);
uint64_t* storeCharacter(uint64_t* s, uint64_t i, uint64_t c);

uint64_t stringLength(uint64_t* s);
void     stringReverse(uint64_t* s);
uint64_t stringCompare(uint64_t* s, uint64_t* t);

uint64_t  atoi(uint64_t* s);
uint64_t* itoa(uint64_t n, uint64_t* s, uint64_t b, uint64_t a, uint64_t p);
```

# A Compiler




- The selfie compiler written in C* translates C* code (self-referential).
- High-level languages have a structure that defines the control flow.
- Machine code has no structure, it is just a sequence of instructions.
- The compiler reads an input program, which is a sequence of characters (ASCII, UTF-8-encoded), and writes machine code.

# The Selfie Library

```
uint64_t leftShift(uint64_t n, uint64_t b);
uint64_t rightShift(uint64_t n, uint64_t b);

uint64_t getBits(uint64_t n, uint64_t i, uint64_t b);
uint64_t getLowWord(uint64_t n);
uint64_t getHighWord(uint64_t n);

uint64_t abs(uint64_t n);
```

```
uint64_t signedLessThan(uint64_t a, uint64_t b);
uint64_t signedDivision(uint64_t a, uint64_t b);

uint64_t isSignedInteger(uint64_t n, uint64_t b);
uint64_t signExtend(uint64_t n, uint64_t b);
uint64_t signShrink(uint64_t n, uint64_t b);

uint64_t  loadCharacter(uint64_t* s, uint64_t i);
uint64_t* storeCharacter(uint64_t* s, uint64_t i, uint64_t c);

uint64_t stringLength(uint64_t* s);
void     stringReverse(uint64_t* s);
uint64_t stringCompare(uint64_t* s, uint64_t* t);

uint64_t  atoi(uint64_t* s);
uint64_t* itoa(uint64_t n, uint64_t* s, uint64_t b, uint64_t a, uint64_t p);
```