

How to print?

<expr> := ... | input | (print <expr>)

Consider:

```
(block
  (print 37)
  (print input))
```

Sketch of compiler code changes

```
| Expr::Id(s) if s == "input" => format!("mov rax, rdi")
| Expr::Print(val) =>
```

```
#[no_mangle]
#[export_name = "\x01snek_print"]
fn snek_print(val : i64) {
  if val == 3 { println!("true"); }
  else if val == 1 { println!("false"); }
  else if val % 2 == 0 { println!("{}", val >> 1); }
  else {
    println!("Unknown value: {}", val);
  }
}

#[no_mangle]
#[export_name = "\x01snek_error"]
fn snek_error(code : i64) { ... }

fn parse_arg(v : &Vec<String>) -> i64 { ... }

fn main() {
  let args: Vec<String> = env::args().collect();
  let input = parse_arg(&args);

  let i : i64 = unsafe { our_code_starts_here(input) };
  print_value(i);
}
```