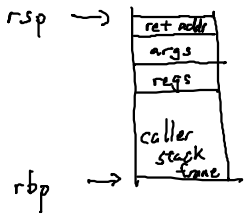
Cardinal

- bird_main : C callee
- stopWithError, printValue : C caller

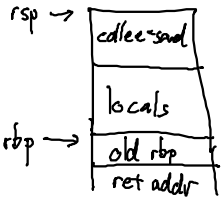
push foo ≡ sub rsp, 8
mov [rsp], foo



x86-64 POSIX C calling conventions

(1) Caller sets up the call and executes callee
  a. save caller-saved regs (volatile) rax, rcx, rdx, r8, r9, r10, r11
  b. set up params: rdi, rsi, rdx, rcx, r8, r9
     rest: onto stack in reverse order
     (arg #7 is at lowest memory addr and on top of stack)
  c. call lbl ≡ push memory addr of next instruction
     jmp lbl

(2) Callee: set up own stack frame
  a. push rbp
  b. allocate local memory
  c. push all callee-saved regs rbx, r12, r13, r14, r15, (rsp, rbp)



(3) Callee runs

(4) Callee: tears down stack frame
  a. pop callee-saved
  b. restoring rbp/rsp (return in rax)
     mov rsp, rbp
     pop rbp
  c. ret ≡ pop rip

(5) a. remove args from stack
  b. pop caller-saved regs

Task:

int foo(int a, int b) asm("foo");

call this function w/ args 8 and 10

```
push r8
mov rdi, 8
mov rsi, 10
call foo
pop r8
```

Dove — declarations

<program> ::= <declaration_list> <expr>

<declaration_list> ::= <declaration> <declaration_list>
| ∈

<declaration> ::= def <identifier>(<param_list>) <expr> end

<param_list> ::= <param>, <param_list>
| <param>
| ∈

<param> ::= <identifier>

<expr> ::= ... | <identifier>(<expr_list>)

```
def dbl(n)
  n*2
end

dbl(6)
```

⟹ 12

```
def triple(k)
  k*3
end
triple(triple(4))
```

⟹ 36

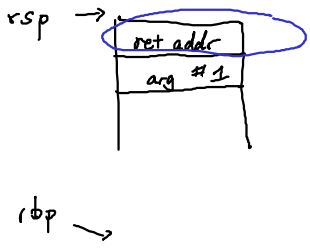
triple(triple(4)) ⟶
triple(4*3) ⟶
triple(12) ⟶
12*3 ⟶ 36

Bird Calling Conventions

same as x86-64 C calling conventions except all args go onto stack

```
def next(n)
  after(n) }  needs 0 bytes
end
```

```
fn_next:
  push rbp
  mov rbp, rsp
  sub rsp, 0
  mov rax, [rbp+16]
  add rax, 2
  mov rsp, rbp
  pop rbp
  ret
```




```
def next(n)
  let k=after(n) in }  needs 8 bytes
  k
end
```