Memory Understanding Activities

```
(+ 2 (read-num))
```

```
entry:
0    mov rax, 8
1    mov [rsp + -8], rax
2    mov [rsp + -16], rdi
3    add rsp, -24
4    call read_num
5    sub rsp, -24
6    mov rdi, [rsp + -16]
7    mov r8, [rsp + -8]
8    add rax, r8
9    ret
```

Registers:

rax

r8

rdi 0

rip 0

rsp 64

Flags:

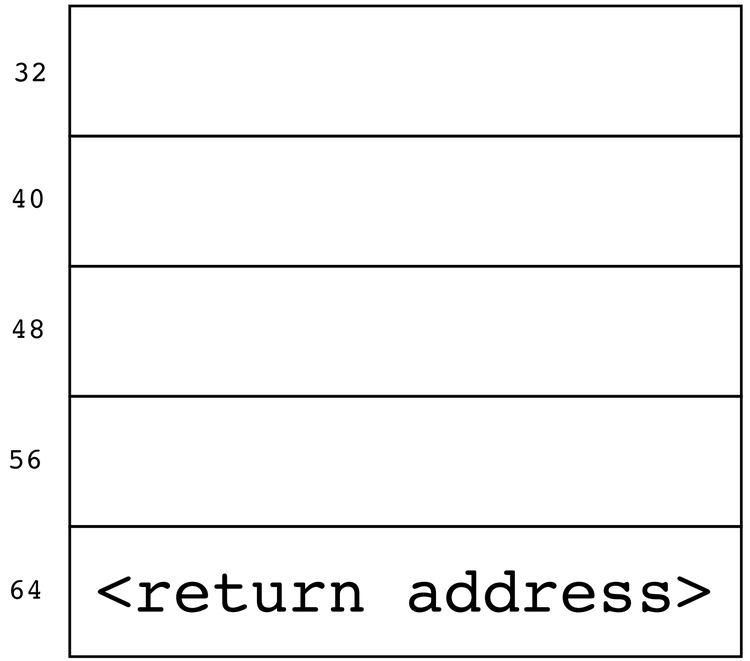
ZF

SF

OF

The heap:

The stack:



When do we subtract 8 from stack_index?

```
(+ 1 2)
```

You might consider this example program.

```
127     | Lst [Sym "+"; e1; e2] ->
128         compile_exp tab stack_index e1
129         @ ensure_num (Reg Rax)
130         @ [Mov (stack_address stack_index, Reg Rax)]
131         @ compile_exp tab (stack_index - 8) e2
132         @ ensure_num (Reg Rax)
133         @ [Mov (Reg R8, stack_address stack_index)]
134         @ [Add (Reg Rax, Reg R8)]
```

When do we add 8 to stack_index?

```
(+ (+ 1 2)(+ 3 4))
```

You might consider this example program.

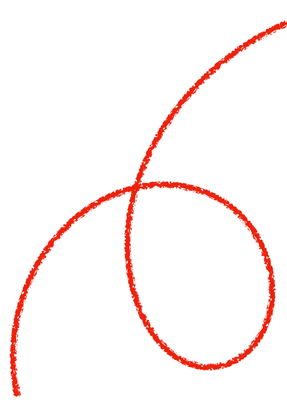
```
127     | Lst [Sym "+"; e1; e2] ->
128         compile_exp tab stack_index e1
129         @ ensure_num (Reg Rax)
130         @ [Mov (stack_address stack_index, Reg Rax)]
131         @ compile_exp tab (stack_index - 8) e2
132         @ ensure_num (Reg Rax)
133         @ [Mov (Reg R8, stack_address stack_index)]
134         @ [Add (Reg Rax, Reg R8)]
```

Why does not knowing the size of a value at compile time mean that we can’t put it on the stack?

```
127     | Lst [Sym "+"; e1; e2] ->
128         compile_exp tab stack_index e1
129         @ ensure_num (Reg Rax)
130         @ [Mov (stack_address stack_index, Reg Rax)]
131         @ compile_exp tab (stack_index - 8) e2
132         @ ensure_num (Reg Rax)
133         @ [Mov (Reg R8, stack_address stack_index)]
134         @ [Add (Reg Rax, Reg R8)]
```

When should we put something on the stack vs. in a register?

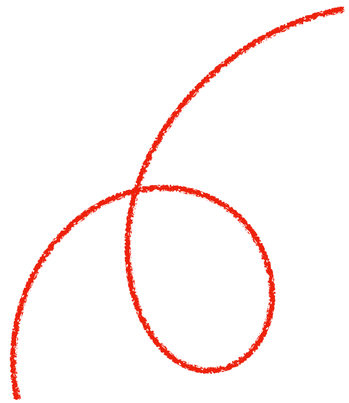
```
(+1(+2(+3(+4(+5(+6(+7(+8(+9(+10(+11(+12(+13(+14(+15(+16
17)))))))))))))))
```



You might consider this example program.

```
127     | Lst [Sym "+"; e1; e2] ->
128         compile_exp tab stack_index e1
129         @ ensure_num (Reg Rax)
130         @ [Mov (stack_address stack_index, Reg Rax)]
131         @ compile_exp tab (stack_index - 8) e2
132         @ ensure_num (Reg Rax)
133         @ [Mov (Reg R8, stack_address stack_index)]
134         @ [Add (Reg Rax, Reg R8)]
```

When does data get removed from the heap?



The only code in our compiler that modifies rdi

```
80      | Lst [Sym "pair"; e1; e2] ->
81        compile_exp tab stack_index e1
82        @ [Mov (stack_address stack_index, Reg Rax)]
83        @ compile_exp tab (stack_index - 8) e2
84        @ [Mov (Reg R8, stack_address stack_index)]
85        @ [
86          Mov (MemOffset (Reg Rdi, Imm 0), Reg R8);
87          Mov (MemOffset (Reg Rdi, Imm 8), Reg Rax);
88          Mov (Reg Rax, Reg Rdi);
89          Or (Reg Rax, Imm pair_tag);
90          Add (Reg Rdi, Imm 16)
91        ]
```