Memory Understanding Activities
ENTRY:

```assembly
entry:
  mov rax, 8
  mov [rsp + -8], rax
  mov [rsp + -16], rdi
  add rsp, -24
  call read_num
  sub rsp, -24
  mov rdi, [rsp + -16]
  mov r8, [rsp + -8]
  add rax, r8
  ret
```

REGISTERS:

- rax
- r8
- rdi 0
- rip 0
- rsp 64

FLAGS:

-ZF (+ 2 (read-num))

THE STACK:

- 32
- 40
- 48
- 56
- 64

<return address>

THE HEAP:

- 0
- 8
- 16
- 32
- 40
- 48
- 56
- 64
When do we subtract 8 from stack_index?

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?

You might consider this example program.

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

```
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?

```plaintext
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?

You might consider this example program.

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

```
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    | ]
```