Section 8: Midterm Review

Four Drill Questions Students Struggled With, Cheat Sheet Walk Through, and Mini-Midterm

CS 164 @ UC Berkeley, Fall 2021
For each expression below, will it result in a **compile-time** error, a **run-time** error, or a **valid** result when run in the **class compiler**?

A. (+ 1 3 4)
B. (+ 1 (let ((x 2)) false))
C. (left (left (pair 1 false)))
D. (if true (* 2 3) (** 2 3))
If our language supported **only booleans and numbers** (no pairs, vectors, refs, etc.) which of the following steps could we **elide** when calling a C function? Explain each.

A. Saving rdi to the stack
B. Bumping rsp to reflect the contents of the stack
C. Ensuring that rsp is a multiple of 16
How many 8-byte heap "cells" will be used when the following program is compiled and executed?

(let ((x false))
  (pair 1
    (if x
      (pair 2 (pair 3 (pair 4 false)))
      (pair 2 (pair 3 false))))
)
The ocaml code below implements a `bool?` operation in the compiler. What should `binary_number_1` be replaced with? Your answer should be a number in binary format (i.e., "0b" followed by some 0s and 1s).

```ocaml
| Lst [Sym "bool?"; arg] →
  |    compile_exp arg
  |      @ [And (Reg Rax, Imm binary_number_1); Cmp (Reg Rax, Imm binary_number_2)]
```
Go Through Cheat Sheet
Bonus: Mini Midterm
(Disclaimer: Not Created By Prof. Chasins)
For each expression below, will it result in a **compile-time** error, a **run-time** error, or a **valid** result when run in the class compiler?

A.  
(\begin{align*}
\text{(let ((v1 (vector 1 true)))} \\
\text{  (let ((v2 (vector 1 true)))} \\
\text{    (vector-get v1 1)))}
\end{align*})

B.  
(\text{vector-length (pair 1 2)})

C.  
(\text{list? (pair 1 2)})

D.  
(\text{vector-get (if (not (char? #\a)) (vector 1 true) (vector 2 true)) 1)}

E.  
(\text{num->char 0})
The OCaml code below implements a **left** projection operation in the compiler. Write **three** s-expressions of the form **(left ...)**. One that will result in a **run time** error, one that will result in a **compile time** error, and one that will result in the **number 22**.

```
| Lst [Sym "left"; e] ->
  compile_exp tab stack_index e
  @ ensure_pair (Reg Rax)
  @ [Mov (Reg Rax, MemOffset (Reg Rax, Imm (-pair_tag)))]
```
What does the **interpreter symbol table** look like at each point in the code?

```
(let ((z false) (x 10)
      do (A_EXP)
       (pair 1
          (if z
            (let (x (+ x 1)) (let (y 9) B_EXP))
            (let (y 8) (C_EXP))
          )
        (x + 1)
        (D_EXP)))))
```

A_EXP: __{z: false; x: 10}______ C_EXP: ___________
B_EXP: _________________________ D_EXP: ___________
What does the **compiler symbol table** look like?
(var stack index in order of appearance)

```
(let ((z false) (x 10)
      do (A_EXP)
      (pair 1
        (if z
          (if (let (x (+ x 1)) (let (y 9) B_EXP))
            (let (y 8) (C_EXP))
            (x + 1)
            (D_EXP))
      (A_EXP: __{z:1; x:2}________               C_EXP: ___________
        B_EXP: ____________________               D_EXP: ___________
      ① ② ③ ④)
```