Compiler Construction
Professor Jeremy Siek

Quiz 4
October 11, 2010

Name: _____

This exam has 4 questions, for a total of 10 points.

1. [2 points] Give the output of the following program:

```
y = 2
x = [1,y]
y = [x,3]
z = y[0]
z[0] = 4
print y is z[1] or x
```

Solution:

```
[4,2]
```

2. [2 points] What is the output of the following program:

```
print ((not ([] + [])) + -1) == False
```

Solution:

```
True
```

3. [3 points] Write down the AST that results from the explicate pass for the following program.

```
print input() + 3
```

Solution:

```
Module(None,
 Stmt(
  [Printnl([
    Let(0_letify,
        CallFunc(Name('input_int'), [], None, None),
        Let(1_letify,
            InjectFrom(int, Const(3)),
            IfExp(Compare(GetTag(Name('0_letify')), [('==', Const(0))]),
                  InjectFrom(int, Add((ProjectTo(int, Name('0_letify')),
                                       ProjectTo(int, Name('1_letify'))))),
                  IfExp(Compare(GetTag(Name('0_letify')), [('==', Const(1))]),
                        InjectFrom(int, Add((ProjectTo(int, Name('0_letify')),
                                             ProjectTo(int, Name('1_letify'))))),
                        InjectFrom(big, CallFunc(Name('add'),
                                          [ProjectTo(big, Name('0_letify')),
                                           ProjectTo(big, Name('1_letify'))],
                                          None, None))
                       )
                 )
            )
       )], None)]))
```

4. [3 points] Write down the x86 assembly code for the program in question 3.

Solution:

```
.globl _main
_main:
        pushl %ebp
        movl %esp, %ebp
        subl $8, %esp
        call input_int
        movl %eax, %ecx
        movl $3, %ebx
        sall $2, %ebx
        orl $0, %ebx
        movl %ecx, %eax
        andl $3, %eax
        cmpl $0, %eax
        sete %al
        movzbl %al, %esi
        cmpl $0, %esi
        je label_23_else
        sarl $2, %ecx
        sarl $2, %ebx
        addl %ebx, %ecx
        sall $2, %ecx
        orl $0, %ecx
        jmp label_24_if_end
label_23_else:
        movl %ecx, %eax
        andl $3, %eax
        cmpl $1, %eax
        sete %al
        movzbl %al, %esi
        cmpl $0, %esi
        je label_21_else
        sarl $2, %ecx
        sarl $2, %ebx
        movl %ecx, %eax
        addl %ebx, %eax
        sall $2, %eax
        orl $0, %eax
        jmp label_22_if_end
label_21_else:
```

```
        movl $3, %esi
        notl %esi
        andl %ecx, %esi
        movl $3, %ecx
        notl %ecx
        andl %ebx, %ecx
        pushl %ecx
        pushl %esi
        call add
        addl $8, %esp
        movl %eax, %ebx
        movl %ebx, %eax
        orl $3, %eax
label_22_if_end:
        movl %eax, %ecx
label_24_if_end:
        pushl %ecx
        call print_any
        addl $4, %esp
        movl $0, %eax
        leave
        ret
```