

Exam 1

Prof. Tak Auyeung

March 3, 2005

Instructions: You may bring any material that is hand-written or printed *prior* to the examination to help you. You can also bring a calculator if you think it may help you. However, you can only use the calculator for numerical computations only.

You, as an individual, are expected to do your own work. This means you cannot seek, receive or otherwise acquire any assistance except clarifications from the professor during an examination. Any communication involving the contents of the subject matter or the examination is considered cheating. Do not initiate or accept such communication, or the result of your examination is automatically voided.

Each correct answer is worth one point, each wrong answer is worth -0.25 point, and each unanswered question is worth zero point.

Make sure you write down you name on the upper right corner *first*, otherwise I cannot give points to anonymous students!

1 Which of the following instructions does not read from memory (other than to get the instruction itself)? Assume all labels are defined correctly.

- A `addl (%ebx),%eax`
- B `movl someVar,%eax`
- C `movl 12,%eax`
- D `movl $12,(%eax)`
- E all of the above do not read from memory

2 Which of the following instruction sequences goes to label L1 if and only if the value of register `eax` is different from the value of register `ebx`?

- A `cmpl %eax, %ebx`
`jz L1`
- B `cmpl %eax, %ebx`
`jnz L1`
- C `cmpl %eax, %ebx`
`jnb L1`
`jna L1`
- D `cmpl %ebx, %eax`
`ja L1`
`jb L1`
- E both 2c and 2d

3 Which flag or flags are interpreted by `jc`?

- A only C
- B only Z
- C only C and Z
- D only O, S and Z
- E only O and S

4 Which flag or flags are set after the following instructions?

```
movb $0x7f, %al
cmpb $0x80, %al
```

- A only S and O
- B only C and S
- C only C, O and S
- D only C and O
- E only C

5 In some cases, the negation of a condition can be handled by changing the comparison operator. Select a pseudocode that is equivalent to the following code:

```
if ( $x < y$ )  $\vee$  ( $y < z$ ) then
    block1
end if
```

- A **if** ($x < y$) \wedge ($y < z$) **then**
 goto L1
end if
block1
L1:
- B **if** ($x \geq y$) **then**
 goto L2
end if
if ($y \geq z$) **then**
 goto L1
end if
L2:
block1
L1:
(p.t.o.)

```

C  if (x < y) then
    goto L2
    end if
    if (y ≥ z) then
        goto L1
    end if
    L2:
    block1
    L1:
D  if (x ≥ y) then
    goto L2
    end if
    if (y < z) then
        goto L1
    end if
    L2:
    block1
    L1:
E  if (x ≥ y) then
    goto L2
    end if
    if (y ≥ z) then
        goto L1
    end if
    L1:
    block1
    L2:

```

6 What is the value of register `eax` at label `test`?

```

.data
someData:
    .long 0x11223344
someOtherData:
    .byte 0xaa, 0xbb, 0xcc, 0xdd
.text
.global _start
_start:
    movl someData,%eax
    movw someOtherData+1,%ax
test:
    nop
    # the rest of the program is not important

```

- A 0x1122bbcc
- B 0x112244aa
- C 0x112233aa
- D 0x1122ccb
- E 0x11aabbcc

7 Which register is modified in the following instruction?

```

cmpl (%ebx, %edx, 1), %eax

```

- A only `%eax`
- B only `%ebx`

- C only `%ebx` and `edx`
- D `eax`, `ebx` and `edx`
- E no register is modified

8 Which flag(s) can also be set if the `C` flag is set after a comparison instruction? Note that these flags may not be set at the same time.

- A only `S` and `O`
- B only `S`
- C only `O`
- D only `C` and `S`
- E `Z`, `O` and `S`

9 What do we know about the following code?

```

L1:
    addl $-1,%eax
    jnz  L1

```

- A this loop always exits, regardless of the value of `eax` before the loop
- B this is an infinite loop (it never exits)
- C this loop exits only if `eax` was odd before the loop
- D this loop exits only if `eax` was even before the loop
- E this loop generates a segmentation fault

10 What is the value of `eax` at label `test`?

```

.data
someData:
    .byte 0x01, 0x02, 0x03, 0x04, 0x05, 0x06
someDataEnd:
.text
.global _start
_start:
    movl $someData, %eax
    movl $0, %ebx
L1:
    cmpl $someDataEnd, %eax
    jz  test
    addb (%eax), %bl
    addl $1,%eax
    jmp L1
test:
    # some more code to follow

```

- A 21
- B `someData`
- C `someDataEnd`
- D this is an infinite loop
- E this code generates a segmentation fault before the loop terminates

11 When we subtract 01101010_2 from 10110111_2 , how many digits result in borrowing from the one to its left?

- A 1
- B 2
- C 3
- D 4
- E 5

12 What is the value of `eax` after the following code?

```
movl $3,%eax
addl %eax,%eax
subl %eax,%eax
addl %eax,%eax
```

- A 0
- B 3
- C 6
- D 9
- E 12

13 What is the value of `edx` at the label `test`?

```
.data
Label1:
    .byte 0x00, 0x01, 0x02, 0x03, 0x04
Label2:
    .word 0x1122, 0x3344, 0x5566
Label3:
    .long 0x50500505
.text
.global _start
_start:
    movl $Label2+1, %eax
    movl $2, %ebx
    movl -3(%eax, %ebx, 2), %edx
test:
    movl $1,%eax
    movl $0,%ebx
    int $0x80
```

- A 0x55663344
- B 0x33445566
- C 0x50505566
- D 0x33441122
- E 0x55334422

14 Which of the following instructions *reads* the `C` flag?

- A `ja`
- B `addl`
- C `movl`
- D `jg`

E `jnz`

15 How do we translate a conditional branch with an exclusive-or in it?

```
if c ⊕ d then
    goto L1
end if
```

Here is a clue. $c \oplus d = (c \wedge \neg d) \vee (\neg c \wedge d)$. Enumerate all four possible cases, and make sure the code goes to label L1 if and only if $c \oplus d$ is true.

- A

```
if c then
    goto L1
end if
if d then
    goto L1
end if
```
- B

```
if c then
    goto L2
end if
if d then
    goto L1
end if
goto L3
L2:
if d then
    goto L3
end if
goto L1
L3:
```
- C

```
if c then
    goto L2
end if
if d then
    goto L1
end if
L2:
```
- D

```
if c then
    goto L2
end if
if d then
    goto L3
end if
goto L1
L2:
if d then
    goto L3
end if
goto L1
L3:
```

E None of the above works