

Write all answers on the answer sheets provided. You may refer to the ASCII chart included with the exam sheet and to the unmodified x86 command summary sheet provided in class.

1. (4 pts) Draw and label the diagram of a CPU we produced in class.
2. (4 pts) Name two significant characteristics of registers.
3. (2 pts) What does it mean to say that a CPU is 32 bits?
4. (3 pts each) Show how each of the values below would be represented in the computer. Write your answers in binary.
  - (a) the 16-bit integer `-47`
  - (b) the 8-bit integer `29`
  - (c) the character `'$'`
  - (d) the single-precision float `-29.875`
5. (3 pts each) Interpret each of the hexadecimal values below as instructed:
  - (a) `0x30` as a character
  - (b) `0xFDF0` as a 16-bit signed integer
  - (c) `0x3E000000` as a single-precision float
6. (3 pts each) Give the 64-, 32-, 16-, and 8-bit names of each of the following registers.
  - (a) `rdi`
  - (b) `rax`
  - (c) `r9`
7. (16 pts) Suppose the following declarations are placed in memory at the address `0x228`. Show the address and contents of each byte generated by the following declarations. Write all answers in hexadecimal. NOTE: Some of these values you already converted to binary in problem 4.

```
section .data
a:  dw  -47
b:  db  29
c:  db  "Cool",10,0
d:  dd  -29.875
e:  db  0x4F
f:  dd  0x524557
```
8. (4 pts) Given the memory configuration of problem 7, what would the following move command accomplish? (Be specific. Show in hexadecimal the value moved.) `mov edi, [b]`

9. (6 pts) Suppose that labels `a`, `b`, and `c` mark locations in memory where 32-bit signed integers are stored. Write the x86 assembly language commands that will perform the following actions:

```
if (c > 100 || c==a) {
    a= a*b;
}
else {
    b= a/c;
}
```

10. (10 pts) The skeleton program below provides the IO macros and gives structure for an assembly language program.

```
----- skeleton.asm -----
#include "iomacros.asm"
#include "dumpregs.asm"

        global main

        section .data

        section .bss

        section .text
main:

theend:

        mov     eax, 60
        xor     rdi, rdi
        syscall
```

Show what declarations and assembly language statements you would add to this skeleton program to use a loop to produce a list of all the values of  $n!$  where  $n = 0, 1, \dots$ . The output should display  $n$  followed by a space followed by  $n!$  followed by a newline character for each value up until an overflow occurs in the calculation. The command `JO` will jump if the Overflow flag is set and `JNO` will jump if the Overflow flag is not set.

NOTE: In case you have forgotten,  $n! = 1 \cdot 2 \cdot 3 \cdots n$  and, by definition,  $0! = 1$ .

11. (4 pts) What commands would you type to compile, link, and run the modified program you created in problem 10?