Name: _____

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing.

1. (15) Give the value (expressed as a decimal number) that will be in EAX after executing

```
movl $25, %eax
andl $-3, %eax
```

Show your work!

2. (15) Say on a 32-bit machine we have C code that includes an **int** array **x**, with 8 columns. Suppose **x[2][3]** has address 0x456. What will be the hex address of **x[3][2]**?

3. (15) Which program in our homework had multiple stacks? **EXPLAIN CAREFULLY**.

4. (15) Fill in the blank with a term from our course for a certain kind of function/subroutine: Using real machines in our course brings large educational benefits, but one advantage of using a simulator is that we could have had homework in which students wrote _____s.

5. (40) Write an assembly language subroutine **countsmall()**, callable from C, with C signature

```
int sep(int *x,int nx)
```

Here **x** is an **int** array of length **nx**, and the function returns a count of the number of elements smaller than **x[0]**.

For instance, the code

```
int y[6] = {5,12,13,2,8,1},m;
m = countsmall(y,6);
```

would result in **m** being 2, while **countsmall(y,5**) would return 1.

Write the full assembly code for **countsmall()**. Be SURE to have comments at the beginning, stating what you are using the various registers for. Assume that NO registers need be saved for the caller. **WRITE LEGIBLY**; write on scratch paper first, and then copy to your exam sheet. (Remember, supplemental papers to your exam sheet are NOT allowed.)

Solutions:

1. 25 is 0x19, i.e. 00011001, and -3 is (in 2 hex digits) 0xfd, i.e. 11111101. And-ing the former with the latter puts a 0 in bit position 1, and leaves the other bits unchanged. But there is already a 0 in that position, so the 25 is unchanged in all bits.

2. From **x[2][3]** to the end of row 2 is a distance of 4 words. Then **x[3][2]** is 3 more words, for a total of 7. Then compute $0x456 + 7 \times 4 = 0x472$.

3. This was the threaded program. Each thread has its own stack.

4. Interrupt service routines.

5.

```
# countsmall() number of elements in an array x that are
# smaller than x[0]

# the C call is
#
#    int countsmall(int *x,int nx)
#
# where nx is the length of x
#
# for instance, the code
#
#    int y[6] = {5,12,13,2,8,1},m;
#    m = countsmall(y,6);
#
# will result in m = 2

.text
.globl countsmall

   # EBX will point to the current element of x
   # ECX will hold x[0]
   # EDX will hold the loop counter
   # EAX will hold the return value

   # assume NO registers need to be saved to protect the caller

countsmall:
   movl 4(%esp), %ebx  # get address of x
   movl 8(%esp), %edx  # get nx
   decl %edx  # skip x[0]
   movl $0, %eax  # count is 0 so far
   movl (%ebx), %ecx  # ECX now holds x[0]
   addl $4, ebx  # skip x[0]
top:
   movl (%ebx), %edi  # EDI now holds x[i]
   cmpl %ecx, %edi
   jge decedx
   incl %eax
decedx:
   addl $4, %ebx
   decl %edx
   jnz top
   ret
```