Instructions You will have eighty minutes to complete the actual open-book, open-note exam. Electronic devices will be allowed only to consult notes or books from local storage; network use will be prohibited. The actual exam will be a little shorter than this practice exam.

For the next four questions, assume the following register and memory state:

<table>
<thead>
<tr>
<th>CPU register</th>
<th>Value (16 bits)</th>
<th>Memory address</th>
<th>Value (8 bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rax</td>
<td>0x0200</td>
<td>0x0300</td>
<td>0x001</td>
</tr>
<tr>
<td>rbx</td>
<td>0x0100</td>
<td>0x0308</td>
<td>0x002</td>
</tr>
<tr>
<td>rcx</td>
<td>0x0080</td>
<td>0x0310</td>
<td>0x030</td>
</tr>
<tr>
<td>rdx</td>
<td>0x0002</td>
<td>0x0318</td>
<td>0x400</td>
</tr>
</tbody>
</table>

1. What is the value of (%rax,%rbx) as a source argument to subq?

2. What is the value of 0x208(%rbx) as a source argument to movq?

3. What is the value of 0x10(%rax,%rcx,2) as a source argument to movq?

4. What is the value of 0x8(%rbx,%rdx,4) as a source argument to leaq?
The next three questions refer to `compare` defined as

```c
int compare(TYPE a, TYPE b) {
    return (VAR1 >= VAR2);
}
```

where the macros `VAR1` and `VAR2` can be defined as `a` or `b` and the macro `TYPE` can be defined as `int`, `long`, or `unsigned`.

5. What definitions of `TYPE`, `VAR1`, and `VAR2` are consistent with the following compiled form?

```asm
xorl %eax, %eax
cmpl %esi, %edi
setge %al
ret
```

6. What definitions of `TYPE`, `VAR1`, and `VAR2` are consistent with the following compiled form?

```asm
xorl %eax, %eax
cmpq %rdi, %rsi
setge %al
ret
```

7. What definitions of `TYPE`, `VAR1`, and `VAR2` are consistent with the following compiled form?

```asm
xorl %eax, %eax
cmpl %edi, %esi
setnb %al
ret
```
8. After running the assembly sequence

```
    orl $0x8, %eax
    movl $0x10, %ebx
    cmpl %ebx, %eax
    jb .L1
    movl $0x8, %eax
.L1:
    movb $3, %cl
    shrl %cl, %eax
```

what is the value of register %eax?
9. Given that the function go

```c
int go(int n, ARGS) {
    int i;
    DECLS
    for (i = 0; i < n; i++)
        a += ((i & 1) ? a : b);
    return a;
}
```

compiles as

```
testl %edi, %edi  # L1
movl %esi, %eax   # L2
jle  L2           # L3
movl %edx, %esi   # L4
xorl %ecx, %ecx   # L5
jmp  L3
L5:
    testb $1, %cl   # L6
    movl %eax, %esi # L7
    cmove %edx, %esi
L3:
    addl $1, %ecx   # L8
    addl %esi, %eax # L9
    cmpl %edi, %ecx # L10
    jne  L5         # L11
L2:
    rep ret
```

pick a combination of ARGS and DECLS (not necessarily in the same row) that fits:

- **ARGS** is int a
- **ARGS** is int b
- **ARGS** is int a, int b
- **DECLS** is empty
- **DECLS** is int a = 0
- **DECLS** is int b = a
The next two questions both use a as defined by

```c
int a[16][32];
```

10. What is the array-access form (i.e., `a[i][j]` for specific `i` and `j`) that is equivalent to `((int *)a)[47]`?

11. What is the pointer-access form (i.e., `((int *)a)[n]` for a specific `n`) equivalent to `a[2][16]`?

12. Given that

```c
int sum_element(int i, int mat1[][M], int j, int mat2[][N]) {
    return mat1[i][j] + mat2[j][i];
}
```

compiles as

```assembly
movslq %edi, %rdi
movslq %edx, %rdx
lea (,%rdi,%rdi,2), %rax
lea (%rsi,%rax,4), %rax
movl (%rax,%rdx,4), %eax
lea (%rdx,%rdx,4), %rsi
lea (%rcx,%rsi,4), %rcx
addl (%rcx,%rdi,4), %eax
ret
```
then what are the values of the constants `M` and `N` among the following possibilities?

- `M = 10` and `N = 6`
- `M = 12` and `N = 2`
- `M = 3` and `N = 5`
- `M = 4` and `N = 5`
- `M = 5` and `N = 12`
- `M = 8` and `N = 17`
The next two questions use fish defined as

```c
typedef struct fish {
    char color[6];
    int variety;
} fish;
```

13. What is `offsetof(fish, variety)`?

14. Given the declaration

```c
fish fa[100];
```

and if the array `fa` starts at address `0x10000`, then what is the address of `fa[32].variety`?
The next two questions refer to the `iterate` function defined as

```java
double iterate(double v, double u, int steps) {
    while (steps--) {
        u = u + v;
        v = 2 * steps;
    }
    return v+u;
}
```

15. Which of the following correctly represents the dependency graph of `iterate` over three iterations, where each column corresponds to a single iteration?

- ![Dependancy Graph 1]
- ![Dependancy Graph 2]
- ![Dependancy Graph 3]
- ![Dependancy Graph 4]
- ![Dependancy Graph 5]

16. Based on the dependency graph, how many cycles will `iterate` take (expressed as a multiple of `steps`)?

7
Answers
1. 0x1
2. 0x2
3. 0x30
4. 0x110
5. TYPE = int, VAR1 = a, VAR2 = b
6. TYPE = long, VAR1 = b, VAR2 = a
7. TYPE = unsigned, VAR1 = b, VAR2 = a
8. 1
9. ARGS is int a, int b and DECLS is empty
10. a[1][15]
11. ((int*)a)[80]
12. M = 3, N = 5
13. 8
14. 0x10188
15. The second one
16. 3*steps