Midterm stats

- Average: 82%
- Median: 83%
- Maximum: 100%
- Minimum: 43%
Recursion in ARM

```
fact:
  CMP r0, #1
  BLE ret_one
  MOV r1, r0
  SUB r0, r0, #1
  BL fact
  MUL r0, r0, r1
  B end

ret_one: MOV r0, #1
end:
  BX lr
```

```
int fact (int n)
{
  if (n <= 1)
    return 1;
  else
    return n * fact(n - 1);
}
```

What is the value returned by fact(1)?

A. One  
B. Two  
C. Three  
D. Six  
E. None of the above
Recursion in ARM

```assembly
fact:
  CMP r0, #1
  BLE ret_one
  MOV r1, r0
  SUB r0, r0, #1
  BL fact
  MUL r0, r0, r1
  B end

ret_one: MOV r0, #1
end:
  BX lr

int fact (int n) {
  if (n <= 1)
    return 1;
  else
    return n * fact(n - 1);
}
```

What is the value returned by fact(2)?
A. One
B. Two
C. Four
D. Six
E. None of the above

Infinite Loop

main: mov r0, #2
  BL fact
  mov r4, r0
Recursion in ARM

```
fact: push {lr}
    CMP r0, #1
    BLE ret_one
    MOV r1, r0
    SUB r0, r0, #1
    BL fact
    MUL r0, r0, r1
end:
    pop {lr}
    BX lr

ret_one: MOV r0, #1
end: pop {lr}
    BX lr
```

```
int fact (int n) {
    if (n <= 1)
        return 1;
    else
        return n * fact(n - 1);
}
```

What is the value returned by fact(3)?

A. One
B. Two
C. Four
D. Six
E. None of the above
Recursion in ARM

fact:

```assembly
push {r1, lr}
CMP r0, #1
BLE ret_one
MOV r4, r0
SUB r0, r0, #1
BL fact
MUL r0, r0, r4
```

```c
int fact (int n) {
    if (n <= 1)
        return 1;
    else
        return n * fact(n - 1);
}
```

What is the value returned by fact(3)?
A. One
B. Two
C. Four
D. Six
E. None of the above
void foo ( ) {
    int arr[3];
    int j = 0;
    for (j = 0; j <= 3; j++)
        *(arr+j) = 0;
}

When the function foo () is called, the program gets stuck in an infinite loop.

- Explain the reason for this behavior.
- Which of the following represents the relative location of the elements of the array and the variable j in memory? Assume memory locations increase left to right

D. Either A or B
E. None of the above
Explaining weird program behavior

void foo () {
    char arr[3];
    int j=0;
    for (j=0; j<=3; j++)
        *(arr+j)=0;
}

Do you think the program would still get stuck in an infinite loop if arr was a char array?
Assume data is aligned and byte ordering is little endian.

A. Yes
B. No

[Diagram of memory alignment and byte ordering]
void foo() {
    char arr[4];
    int j = 0;
    for (j = 0; j <= 4; j++)
        *(arr+j) = 0;
}

Do you think the program would still get stuck in an infinite loop for the new code?
Assume data is aligned, byte ordering is Little Endian.
Think about why or why not
A. Yes
B. No
HAPPY THANKSGIVING!!