This quiz has 1 questions, for a total of 10 points.

1. 10 points Perform register allocation and convert the following C function to x86. Show your work for each phase of the register allocation process, including liveness analysis, interference and move graph building, graph coloring, spill code generation, and removing trivial moves.

```c
int f(int a, int b) {
    int d = 0;
    int e = a;
    do {
        d += b;
        e -= 1;
    } while (e);
    return d;
}
```

**Solution:**

Liveness analysis:

```c
int f(int a, int b) {
    int d = 0; // live: {a,b}
    int e = a; // live: {a,b,d}
    do {
        d += b; // live: {b,d,e}
        e -= 1; // live: {b,d,e}
    } while (e); // live: {b,d,e}
    return d; // live: {d}
}
```

Interference graph:

```
  a  
 /   
 b  
  
  d  
  
  e  
```

Move graph:

```
  a  
 /   
 b  
  
  d  
  
  e  
```

Coloring:

- a: 8(%ebp)
- b: 12(%ebp)
- d: %eax
- e: %ecx

(It is also OK to move ‘a’ and ‘b’ into local variables that get assigned to registers.)
### x86 code:

```assembly
f:
pushl %ebp
movl %esp, %ebp
movl $0, %eax
movl 8(%ebp), %ecx
L2:
    addl 12(%ebp), %eax
    subl $1, %ecx
    cmpl $0, %ecx
    jne L2
popl %ebp
ret
```