



COMP - Intro to Compilers II (MIEIC - Compilers - 2021)

131

Responses

4.3

Average Score

Closed

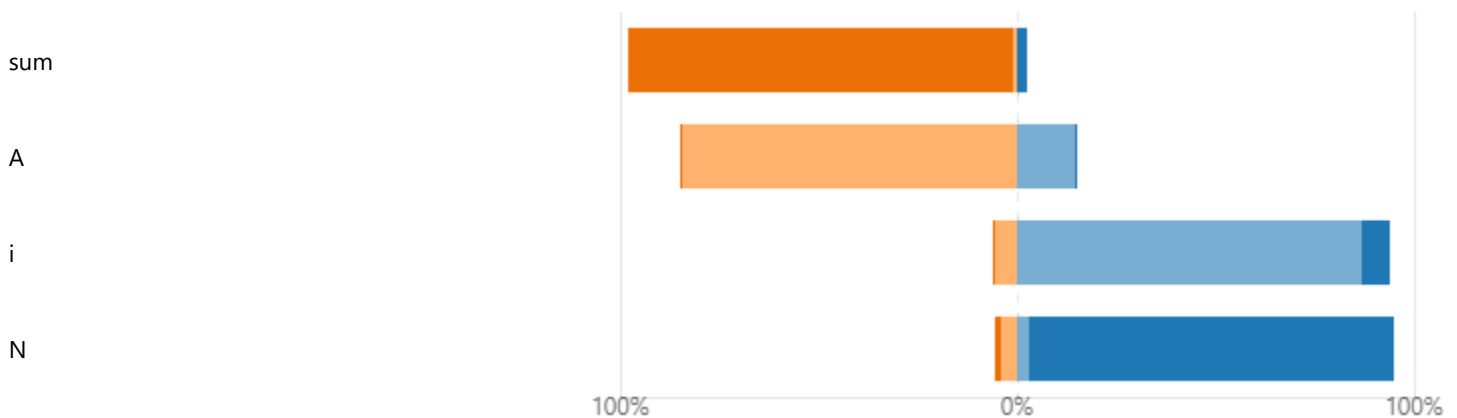
Status

1. Consider the following C function and the output assembly code by gcc -O0 for MIPS:

```
int sum(int A[], int N) { int i, sum = 0; for(i=0; i<N; i++) { sum = sum + A[i]; }  
return sum; } //sum(int*, int):  
1. blez $5,$L4 2. sll $5,$5,2 3. addu $5,$4,$5  
4. move $2,$0 $L3: 5. lw $3,0($4) 6. addiu $4,$4,4 7. bne $5,$4,$L3  
8. addu $2,$2,$3 9. j $31 9. nop $L4: 10. j $31 11. move $2,$0
```

2. Indicate the register the compiler assigned to each variable: (1 point)

■ \$2 ■ \$3 ■ \$4 ■ \$5



3. Why did the compiler output an instruction after each j (jump) instruction? (1 point)

86% of respondents (113 of 131) answered this question correctly.

● because the target MIPS proc... 113 ✓

● because the branch may not b... 18



4. The compiler optimized the code considering that the loop executes at least 1 iteration (1 point)

85% of respondents (112 of 131) answered this question correctly.

● TRUE 19

● FALSE 112 ✓



5. The implementation of the FOR loop is similar to (1 point)

76% of respondents (99 of 131) answered this question correctly.

● a do...while 99 ✓

● a while 32



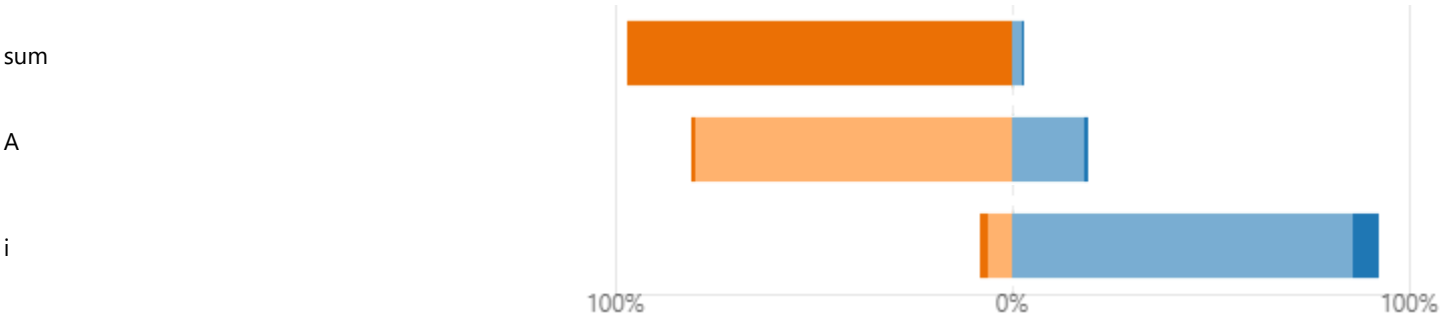
6. Consider the following C function and the output assembly code by gcc -O0 for MIPS:

```
#define N 1000
int sum(int A[]) {
    int i, sum = 0;
    for(i=0; i<N; i++) {
        sum = sum + A[i];
    }
    return sum;
}

//sum(int*):
1.    addiu $5,$4,4000
2.    move  $2,$0 $L2:
3.    lw    $3,0($4)
4.    addiu $4,$4,4
5.    bne   $5,$4,$L2
6.    addu  $2,$2,$3
7.    j     $31
8.    nop
```

7. Indicate the register the compiler assigned to each variable: (1 point)

\$2 \$3 \$4 \$5



8. The compiler optimized the code considering that the loop executes at least 1 iteration (1 point)
94% of respondents (123 of 131) answered this question correctly.

TRUE 123 ✓
FALSE 8



9. The implementation of the FOR loop is similar to (1 point)
88% of respondents (115 of 131) answered this question correctly.

a do...while 115 ✓
a while 16

