# CSPB 2400 - Park - Computer Systems

<u>Dashboard</u> / My courses / <u>2241:CSPB 2400</u> / <u>5 February - 11 February</u> / <u>Reading quiz on CS:APP 3.6</u>

State Finished

Completed on Saturday, 10 February 2024, 9:11 PM

Time taken 14 mins 34 secs

Marks 12.00/12.00

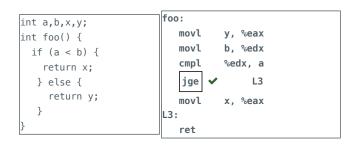
Grade 10.00 out of 10.00 (100%)

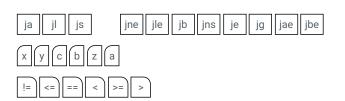
Question 1

Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.





Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
foo:
int a,b,x,y;
                                      y, %eax
                              movl
int foo() {
                                      b, %edx
                              movl
 if (a == b) {
                              cmpl
                                      %edx, a
   return x;
                              jne
                                           L3
  } else {
     return y;
                             movl
                                      x, %eax
  }
                           L3:
                              ret
```



Your answer is correct.

## Question 3

Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
int a,b,x,y;
                               foo:
int foo() {
                                  movl
                                          y, %eax
 if (a <=
               b) {
                                  movl
                                          b, %edx
                                  cmpl
                                          %edx, a
   return x;
                                  jg
                                         L3
  } else {
                                  movl
                                          x, %eax
     return y;
                               L3:
  }
```



Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
unsigned int a,b,'
                              foo:
int x,y;
                                         y, %eax
                                 movl
int foo() {
                                         b, %edx
                                 movl
 if (a >= b) {
                                 cmpl
                                         %edx, a
   return x;
                                  jb
                                               L3
  } else {
                                 movl
                                         x, %eax
     return y;
                              L3:
  }
                                 ret
```



Your answer is correct.

## Question 5

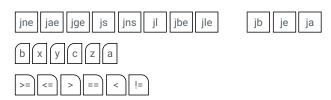
Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
int foo(a) {
    if (a <= 5) {
        return -10;
    }
    else {
        return 10;
    }
}</pre>
foo:
cmpl $0x5, %edi

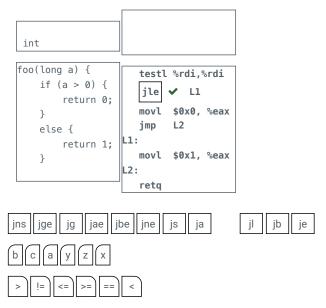
jg ✔ L1
mov $0xffffffff6, %eax
jmp L2
L1:
mov $0xa, %eax
L2:
retq
```



Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.



Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
foo:
                                    movl
                                            $0, i
                                    jmp
                                            .L2
                            .L3:
int sum = 2;
                                    movl
                                            sum, %eax
int i;
                                    addl
                                            $3, %eax
int foo() {
                                    movl
                                            %eax, sum
  i = 0;
                                    movl
                                            i, %eax
  while (i < 5){
                                    incl
                                            %eax
     sum = sum + 3;
                                    movl
                                            %eax, i
      i++;
                            .L2:
  }
                                            i, %eax
                                    movl
return sum;
                                             $4
                                    cmpl
                                            . L3
                                    jle
                                    movl
                                            sum, %eax
                                    ret
```

\$3 \$2 \$5 \$6 \$1

Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
foo:
                                             movl
                                                      $0, i
                                             jmp
                                                      .L2
int sum = 2;
                                      .L3:
int i;
                                             movl
                                                     sum, %eax
int foo() {
                                                     $3, %eax
                                             addl
                                             movl
                                                      %eax, sum
for (i = 0; i < 5; i++)
                                                      i, %eax
                                             addl
                                                     $1, %eax
   sum = sum + 3;
                                             movl
                                                      %eax, i
                                      .L2:
                                                      i, %eax
                                             movl
return sum;
                                                      $4
                                             cmpl
                                             jle
                                                      .L3
                                             movl
                                                      sum, %eax
```

\$3 \$6 \$5 \$1 \$2

Correct

Mark 1.00 out of 1.00

The assembly code on the right partially implements the C function shown on the left. Fill in the missing instruction to correctly implement the C function on the left.

```
foo:
                                                  movl
                                                           $0, i
                                                  jmp
                                                           .L2
int sum = 2;
                                          .L3:
int i;
                                                          sum, %eax
                                                  movl
int foo() {
                                                  addl
                                                          $3, %eax
                                                  movl
                                                          %eax, sum
for (i = 0; i < 3) \checkmark ; i++)
                                                  movl
                                                          i, %eax
                                                          $1, %eax
                                                  addl
   sum = sum + 3;
                                                  movl
                                                          %eax, i
                                          .L2:
                                                          i, %eax
                                                  movl
return sum;
                                                  cmpl
                                                           $2, %eax
                                                  jle
                                                           .L3
                                                           sum, %eax
                                                  movl
```

6 5 1 2 4

Your answer is correct.

### Ouestion 10

Correct

Mark 1.00 out of 1.00

In a C function, we have switch statement with four cases.

Given the assembly code and jump table below, what is the value of the largest **case** in the switch statement? (enter the value in base-10 format).

```
movq 8(%rbp), %rax
subq $15, %rax
cmpq $3, %rax
ja .L2
jmp .L6(, %rax,4)

.L6
.quad .L3
.quad .L2
.quad .L2
.quad .L5
```

Correct

Mark 1.00 out of 1.00

Write a C function **func** that performs the actions of the following assembly code:

# For example:

Test	Result	
test(5,4);	0K1	

# Answer: (penalty regime: 0 %)

```
1  int func(int a, int b) {
    if (a > b) {
        return 0;
    }
    else {
        return 1;
    }
}
```

	Test	Expected	Got	
~	test(5,4);	0K1	0K1	~
~	<pre>test(rand(), rand());</pre>	0K2	0K2	~
<b>~</b>	<pre>test(rand(), rand());</pre>	0K3	0K3	~

## Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Correct

Mark 1.00 out of 1.00

Write a C function func that performs the actions of the following assembly code. Your function must be named func.

The function takes four arguments passed in registers %edi, %esi, %edx and %rcx.

The function returns a single argument in %eax.

```
func:
       movl
                %edi, %eax
.L3:
       movslq %eax, %rdi
        cmpl
                %esi, (%rcx,%rdi,4)
        jе
                .L2
        addl
                $1, %eax
        cmpl
                %edx, %eax
        jl
                .L3
        movl
                $-1, %eax
.L2:
        ret
```

#### For example:

Test	Result
test(1,target1);	0K1

Answer: (penalty regime: 10, 20, ... %)

	Test	Expected	Got	
~	test(1,target1);	0K1	0K1	~
~	test(2,target2);	0K2	0K2	~
~	test(3,target3);	0K3	0K3	~

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.