

# 15-213 Introduction to Computer Systems

## Exam 1

February 27, 2007

Name:

Andrew User ID:

Recitation Section: \_\_\_\_\_

- This is an open-book exam. Notes are permitted, but not computers.
- Write your answer legibly in the space provided.
- You have 80 minutes for this exam.

# Assignment Project Exam Help

<https://eduassistpro.github.io/>  
Floating Point  
Add WeChat edu\_assist\_pro

	Problem	Max	Score
Assembly Language			
Calling Conventions	4	10	
Structures and Alignment	5	10	
Out-of-Order Execution	6	15	
	<b>Total</b>	<b>75</b>	

## 1. Integers (10 points)

For each of the following propositions, write in **all** comparisons that make it **always** true among the four possibilities:

< > == !=

If none are guaranteed to hold, please indicate that explicitly by marking it with an **X**. We have filled in the first two for you as examples. Assume the variables are declared with

```
int x,y;
```

and initialized to some unknown values. You may assume that `int`'s are 32 bits wide, `char`'s are 8 bits wide and that right shift is arithmetical on signed numbers and logical on unsigned numbers.

If $x > y$ then $y$	$\left\{ \begin{array}{l} < \\ & != \end{array} \right\}$	$x$
If $x < 0$ then $x+1$	$\left\{ \begin{array}{l} & X \end{array} \right\}$	0
If $x > 0$ then	$\left\{ \begin{array}{l} & \end{array} \right\}$	
If $(x \gg 31) < 0$ then $x$	$\left\{ \begin{array}{l} & \end{array} \right\}$	
If $((x \ll 31) \gg 31) < 0$ then $x \& 1$	$\left\{ \begin{array}{l} & \end{array} \right\}$	0
If $x > y$ then $(\text{unsigned})x$	$\left\{ \begin{array}{l} & \end{array} \right\}$	$y$
If $((\text{unsigned char})x \gg 1) < 64$ then $(\text{char})x$	$\left\{ \begin{array}{l} & \end{array} \right\}$	0

## 2. Floating Point (15 points)

- (10 pts) Fill in the blank entries in the following table. We assume an IEEE representation of floating point numbers with 1 sign bit,  $k = 3$  bits for the exponent and  $n = 4$  bits for the fractional value. This means the bias is  $2^{3-1} - 1 = 3$ .

We are only considering positive numbers.

Value	Form $M \times 2^E$ for $1 \leq M < 2$	Hexadecimal representation	Decimal fraction
Smallest denorm.		0x01	
Smallest norm.			
One	$1.0 \times 2^0$		1
Largest norm.			
Infinity	XXXXX		XXXXX

- (5 pts) With standard IEEE representation with 1 sign bit,  $k = 8$  bits for the exponent, and  $n = 23$  bits for the fractional value, what is the smallest positive value for  $n$  declared with `int n;`

```
(int)(float)n != n;
```

### 3. Assembly Language (15 points)

Consider the following recursive C program that sorts a segment of an integer array into ascending order in place.

```
/* qsort(A, low, high) sorts subarray A[low]..A[high] */
void qsort (int* A, int low, int high) {
    int pivot, i, k;
    if (low >= high) return;
    pivot = A[high];
    i = low;
    k = high;
    while (i < k) {
        if (A[i] < pivot) {
            i++;
        } else {
            A[k] = A[i];
            k--;
            A[i] = A[k];
        }
    }
    A[k] = pivot;
    qsort(A, low, k-1);
    qsort(A, k+1, high);
}
```

The while loop is compiled to the following assembly instructions. **Hint:** The entry point of the loop is at label .L8. **Hint:** Do not add instructions until you have answered questions 1 and 2 on the next page.

```
.L13:
    incl    %edx
    jmp     .L8

.L7:
    movslq  %edx, %rcx
    movl    (%rbp,%rcx,4), %eax
    cmpl    %r8d, %eax
    jl      .L13
    movl    %eax, (%rbp,%rdi,4)
    decl    %ebx

    _____ <first missing instruction>

    _____ <second missing instruction>
    movl    %eax, (%rbp,%rcx,4)

.L8:
    cmpl    %ebx, %edx
    jl      .L7
```

1. (5 pts) Complete the following table associating C variable or expressions with registers.

C Expression	Register
pivot	%r8d
A	
i	
k	
(long)k	
(long)i	

2. (5 pts) The register %eax holds temporary values during the loop computation. List all the expressions in the original C program whose values it holds.

3. (5 pts) Fill in the two missing instructions.

**Assignment Project Exam Help**

**<https://eduassistpro.github.io/>**

**Add WeChat edu\_assist\_pro**

#### 4. Calling Conventions (10 points)

After the code of the while loop shown before, we find the following instructions to implement the first recursive call.

```
movl    %r8d, (%rbp,%rdi,4)
leal    -1(%rbx), %edx
movq    %rbp, %rdi
call    qsort
```

1. (5 pts) By the x86-64 calling conventions, which of the mentioned registers are guaranteed to have the same value when `qsort` returns as right before the call? Write **yes** or **no** into the space.

Register	Same?
%r8d	
%rbp	
%rdi	
%rbx	
%edx	

Assignment Project Exam Help

We are surprised that following the first recursive call, we find no further recursive call, but instead the inst

```
leal    1(%rbx), %
jmp     .L14
```

<https://eduassistpro.github.io/>

where `.L14` is a label near the beginning of the compiler used an optimization to eliminate the second recursive call in favor of a loop.

Add WeChat edu\_assist\_pro

2. (5 pts) Complete the following rendering of the compiler's optimization in C in the form of a new while loop.

```
void qsort (int* A, int low, int high) {
    int pivot, i, k;
    /* eliminated here: if (low >= high) return; */
    while (_____) {
        pivot = A[high];
        i = low;
        k = high;
        while (i < k) { ... as before ... }
        A[k] = pivot;
        qsort(A, low, k-1);
        _____ /* was: qsort(A, k+1, high); */
    }
}
```

## 5. Structures and Alignment (10 points)

This question concerns alignment on an x86-64 architecture and pointer arithmetic. Consider the following C structure declaration.

```
struct box {
    int tag; /* 0 = int, 1 = long, 2 = float, 3 = double */
    union {
        int i;
        long l;
        float f;
        double d;
    } data;
};
```

1. (3 pts) What is the total size of a box structure on an x86-64, expressed in bytes?

## Assignment Project Exam Help

2. (2 pts) An element of type `struct box` must be aligned at 0 modulo \_\_\_\_\_.

3. (5 pts) Write a C function that takes the high 4 bytes of a long as an argument and returns the value of the tag field. Use a box structure and point to the tag field. `get_tag` should return the tag value, or 0 if the argument is 0. `get_tag` should ignore the low 4 bytes of the argument.

<https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro

## 6. Out-of-Order Execution (15 points)

We now return to the earlier quicksort procedure. If the array is already sorted, the first conditional branch instruction in the code fragment below will always be taken.

1. (5 pts) Assume that the processor correctly predicts this, as well as the second branching instruction (for example, because they both go backwards). On the right-hand side, show the translation of the assembly language instructions into execution unit operations with register renaming. Do not rename registers that are invariant throughout multiple loop iterations. We have already filled in the translations of the conditional branches; the unconditional jump is handled by the instruction fetch unit. Consider .L7 as the starting point of an iteration.

```
.L13:                                |
      incl    %edx                  |
      jmp     .L8                    | (handled in fetch unit)
.L7:                                |
      movslq  %edx, %rcx            |
      movl    (%rbp,%rcx,4), %eax   |
      cmpl    %rcd, %eax            |
      jl      .L13                  | jl-taken cc.1a
      <om
.L8:                                |
      cmp     %eb                   |
      jl      .L7                    | jl-taken cc.1b
```

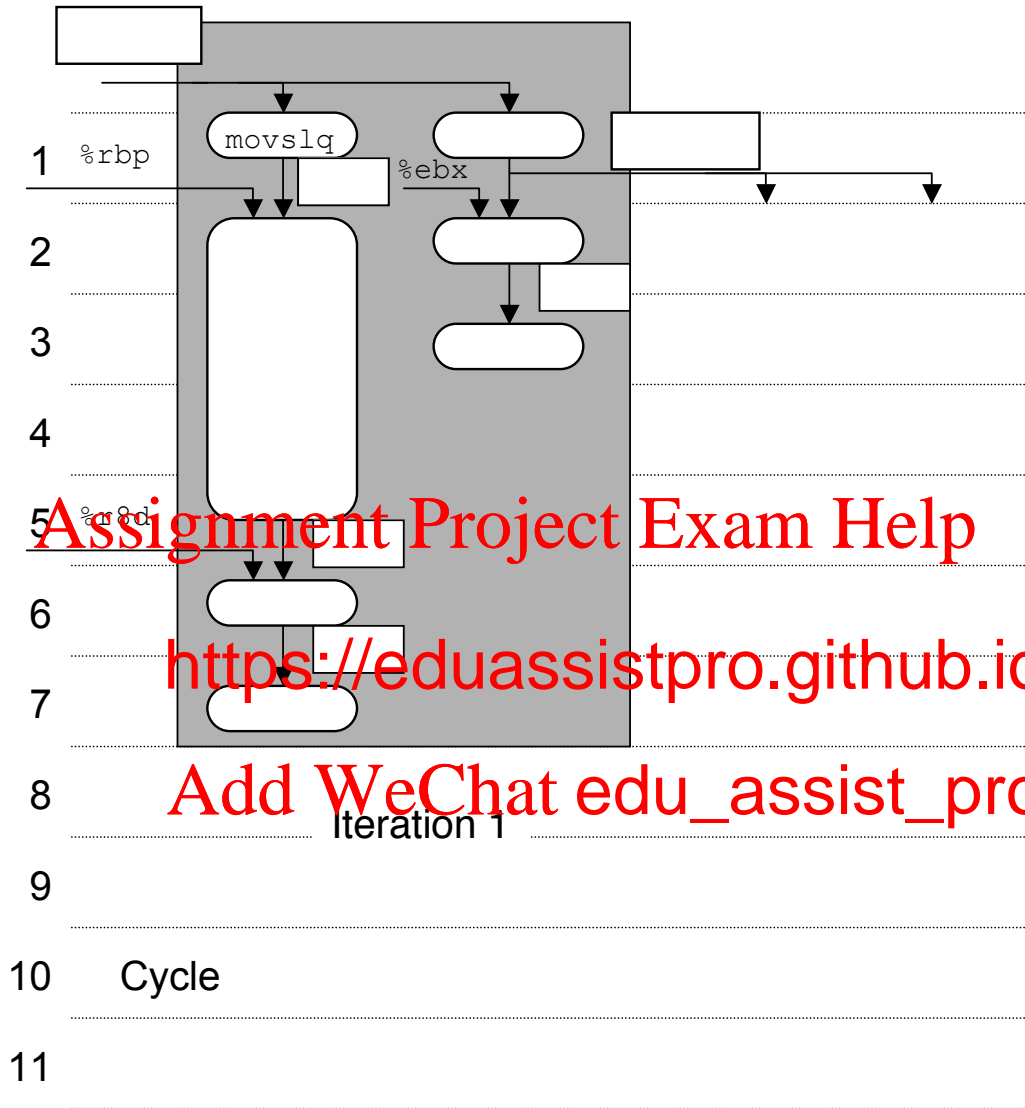
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



2. (7 pts) Complete the labeling in the following timed data dependency diagram. We have already filled in the registers that are not renamed and one register move operation.



3. (3 pts) What is the theoretical CPE for this loop, assuming perfect branch prediction and no resource limitations?

## Worksheet

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## Worksheet

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## Worksheet

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro