

Bits and Bytes/Integers
Floating Point
Machine Prog: Basics

姓名：

学号：

答卷说明：

- a. 答卷前，考生务必将自己的姓名填写在试卷指定位置。
- b. 答题时，请将答案填写在试卷和答题卡相应位置。如需改动，请用签字笔将原答案划去，再在规定位置填写修正后的答案。未在规定区域作答的答案无效。
- c. 本卷共4页，卷面分110分。考试结束后，试卷由助教统一收回。
- d. 如出现中英文标点混用的情况属不严谨，请大家视作英文标点作答。

一、选择题(60分) 每题只有一个正确答案

- () 1. 对于x86-64处理器，下列关于比特数的说法中正确的是____。
A. 指针可指向的内存地址必须是64比特对齐的。
B. 通用寄存器 (e. g. %rax) 大小为64比特。
C. char*类型的变量大小为8比特。
D. long类型的变量大小为32比特。
- () 2. 下面关于大端法的叙述，正确的是____。
A. 使用Intel兼容处理器的IBM个人计算机使用大端法。
B. 大端法指的是低地址存放低字节。
C. 通常情况下，网络传输使用大端法。
D. 对于使用大端法的处理器，若把十六进制数0x01234567存储在地址0x100~0x103中，则0x100地址存储的数字是0x67。
- () 3. 下面关于布尔代数的叙述，**错误**的是____。
A. 设x, y, z是整型，则 $x \wedge y \wedge z == y \wedge z \wedge x$ 。
B. 任何逻辑运算都可以由与运算(&)和异或运算(^)组合得到。
C. 设m, n是char*类型的指针，则下面三条语句
“ $*n = *m \wedge *n; *m = *m \wedge *n; *n = *m \wedge *n;$ ”可以交换*m和*n的值。
D. 已知a, b是整型，且 $a + b + 1 == 0$ 为真。则 $a \wedge b + 1 == 0$ 为真。
- () 4. 下面关于整数运算的叙述，**错误**的是____。
A. 设m, n是整型，则可以使用公式 $(m * n) / m == n$ 来判断m*n是否溢出。
B. 设x, k (k=0, 1, 2, ..., 30)是整型，则表达式 $x / (1 \ll k) == (x < 0 ? x + (1 \ll k) - 1 : x) \gg k$ 为真。
C. 设a, b是整型，则 $a * b == (\text{int}) ((\text{unsigned}) a * b)$ 为真。
D. 两个大于0的整型求和后如果大于零，那么该求和操作没有溢出。

()5. 假设有下面 x 和 y 的程序定义

```
int x = a >> 2;
```

```
int y = (x + a) / 4;
```

那么有____个位于闭区间 $[-8, 8]$ 的整数 a 能使得 x 和 y 相等.

A. 12 B. 13 C. 14 D. 15

()6. 下面关于IEEE754标准下浮点数的叙述, 正确的是____ (提示: 不必从语言学角度考察各个选项的正确性).

A. 浮点乘法不满足交换律、结合律和分配律.

B. 指数位全为0的浮点数一定是非规格化数.

C. 设浮点数的小数字段描述的小数值为 f ($0 \leq f < 1$). 因为我们总能调整阶码使得尾数 M 在范围 $1 \leq M < 2$ 之中(假设没有溢出), 所以我们将尾数 M 定义为 $1+f$, 以此多获得一个精度位.

D. 设 x, y 为double类型的无穷大(inf), 则 $x==y$, $x+y==\text{inf}$, $x-y==\text{inf}$ 都为真.

()7. 下面不是 X86-64正确的寻址方式的指令是____.

A. `movl $34, (%eax)`

B. `movl (%eax), %eax`

C. `movl $23, 10(%edx, %eax)`

D. `movl (%eax), 8(%ebx)`

()8. 在x86-64下, 以下说法错误的是____.

A. 栈指针`%rbp`用于指明运行时栈的结束位置.

B. `movl` 指令以寄存器作为目的时, 会将该寄存器的高位 4 字节设置为0.

C. `cltq` 指令的作用是将`%eax`符号扩展到`%rax`.

D. `movabsq` 指令只能以寄存器作为目的.

()9. 在32位机器下, 假设有如下定义`int array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}`; 某一时刻, `%ecx`存着第一个元素的地址, `%ebx`值为3, 那么下列操作中____将`array[3]`移入了`%eax`.

A. `leal 12(%ecx), %eax`

B. `leal (%ecx, %ebx, 4), %eax`

C. `movl (%ecx, %ebx, 4), %eax`

D. `movl 8(%ecx, %ebx, 2), %eax`

()10. 有如下结构定义和程序片段

```
struct A{
```

```
    char c; int i; double d; int array[10];
```

```
};
```

```

struct B{
    int array[10]; double d; char c; int i;
};
void foo(struct A *pa, struct B *pb, int index){
    pb->i = pa->array[index];
}

```

在 Linux 下使用 GCC 编译器, 仅采用 -O2 选项, 上述代码对应的汇编语言是: (将选项内的数依次填入横线上)

```

movslq %edx, %rdx
movl ____(%rdi,%rdx,____), %eax
movl %eax, ____(%rsi)

```

A. (16, 4, 52) B. (24, 4, 52) C. (16, 4, 49) D. (24, 4, 49)

二、非选择题(50分) 请将答案填写在试卷相应位置

11. (25分) 考虑有一种基于IEEE浮点格式的9位浮点表示格式A. 现在已知 $-\frac{9}{16}$ 的位模式可以表示为“101100010”, 则1.0使用格式A可以表示为 _____ (用二进制数表示); $\frac{21}{64}$ 使用格式A可以表示为 _____ (用二进制数表示); 该格式可以表示的最大非规格化数为 _____ (用分数表示); 该格式可以表示的最小规格化数为 _____ (用十六进制表示). 假设格式 A 变为 1 个符号位, k+1 个阶码位, n-1 个小数位, 那么能表示的实数数量会 _____ (填“增加”, “降低”, “不变”之一).

12. (20分) 假设%rax、%rbx的初始值都是0. 请据汇编代码将???填写完整.
 (1) _____. (2) _____. (3) _____. (4) _____.

	%rax	%rbx
movabsq \$0x0123456789ABCDEF, %rax		
---->	0x0123456789ABCDEF	0x0000000000000000
movw %ax, %bx		
---->	0x0123456789ABCDEF	(1) ????????????????
movswq %bx, %rbx		
---->	0x0123456789ABCDEF	(2) ????????????????
movl %ebx, %eax		
---->	(3) ????????????????	(2) ????????????????
Movabsq \$0x123456789ABCDEF, %rax		
---->	0x0123456789ABCDEF	(2) ????????????????
cltq		
---->	(4) ????????????????	(2) ????????????????

13. (5分) 将下列汇编代码翻译成C代码, 横线处应该填写_____.

<pre>func: 1 movq %rdi, %rax 2 salq \$4, %rax 3 subq %rdi, %rax 4 movq %rax, %rdi 5 leaq 0(,%rsi,8), %rax 6 subq %rsi, %rax 7 addq %rdi, %rax 8 ret</pre>	<pre>// a in %rdi, b in %rsi long func(long a, long b) { return _____; }</pre>
--	--

2021秋ICS小班18班第（2）次考试
答题卡

一、选择题部分

1	2	3	4	5	6	7	8	9	10

二、非选择题部分

11.

12.

13.

参考答案

一、选择题 每题6分, 共60分.

1-5: B C D A B

6-10: B D A C A

二、非选择题 每空5分, 共50分.

11. 001110000 101010101 15/1024 0x1ef 增加

12. 0x0000000000000CDEF 0xFFFFFFFFFFFFCDEF

0x00000000FFFFCDEF 0xFFFFFFFF89ABCDEF

13. $a*15+b*7$