

1-1 判断:  $x$  是 int,  $x \gg 3 == x/8$

1-2 判断:  $x$  是 int,  $x == (\text{int})(\text{float})x$

1-3 判断:  $-1 < 0U$

1-4 判断:  $x$  是 int,  $(x|-x) \gg 31 == -1$

1-5 判断: `cmpq (%rax),6(%rsp)`

1-6 判断: `lea %r12,-x2006ae(%rip)`

2-1 已知变量  $x$  的类型为 int, 位于地址  $0x100$  处, 它的十六进值为  $0x01234567$ , 在采用小端模式的机器中, 地址范围  $0x100 \sim 0x103$  存储的字节依次为

- A.  $0x67$   $0x45$   $0x23$   $0x01$       B.  $0x45$   $0x67$   $0x01$   $0x23$   
C.  $0x01$   $0x23$   $0x45$   $0x67$       D.  $0x23$   $0x01$   $0x67$   $0x45$

2-2 已知 IEEE754 单精度浮点数的值为  $0xC0B00000$ , 那它的十进制为

- A. -2.75      B. -0.75      C. -1.5      D. -5.5

2-3 用 gcc 编译 C 语言程序生成在 x86-64 上运行的 64 位程序, 给数据类型 int、float、long、char \* 分配的字节数分别是

- A. 4、4、8、8  
B. 4、8、4、8  
C. 4、8、4、1  
D. 4、2、8、1

2-4 已知 C 程序段如下, 请问 for 语句的循环次数为

```
#define DELTA sizeof(int)
int i;
for (i = 10; i-DELTA >= 0; i-- DELTA)
```

- A. 7      B. 5      C. 死循环      D. 6

2-5 假设  $x$  和  $y$  的字节值分别为  $0x66$  和  $0x39$ , 则表达式  $x \& y$  的结果为

- A. 1      B.  $0x7F$       C. 0      D.  $0x20$

2-6 在 x86-64 机器上, TMax 表示的值为

- A.  $2^{63}-1$       B.  $2^{64}-1$       C.  $2^{63}$       D.  $2^{64}$

2-7 采用 GCC 将下述 C 程序编译成汇编代码

```
void multstore(long x, long y, long *dest) {
    long t = mult2(x, y);
    *dest = t;
}
```

变量 x, y, dest 参数将分别采用下列哪个选项中的寄存器进行传递

- A.rsi, rdx, rdi
- B.rdx, rdi, rsi
- C.rsi, rax, rcx
- D.rdi, rsi, rdx

2-8 C 语言有如下声明

```
int array[6] = {10, 20, 30, 40, 50, 60}
```

假设编译器把变量 array 放到 %rcx 寄存器中, 且已知 %rbx 的值为 2, 把 array[2] 的值传送到 %edx 寄存器中的指令是

- A.movl (%rbx, %rcx, 4), %edx
- B.movl (%rcx, %rbx, 4), %edx
- C.leal (%rbx, %rcx, 4), %edx
- D.leal (%rcx, %rbx, 4), %edx

2-9 在 Linux 下, 下列哪个命令行可以将 C 程序 test.c 编译生成可执行文件 ( )

- A.gcc -O1 -o result test.o
- B.objdump -d test.o
- C.gcc -Og -c test test.c
- D.gcc -Og -o result test.c

2-10 表示传送字节的是下述哪条指令

- A.movw    B.movb    C.movl    D.movq

2-11 16 位补码整数所能表示的范围是 ( )

- A. $-(2^{16}-1) \sim +(2^{16}-1)$
- B. $-2^{15} \sim +(2^{15}-1)$
- C. $-2^{16} \sim +(2^{16}-1)$
- D. $-(2^{15}-1) \sim +(2^{15}-1)$

2-12 用 gcc 编译下面 C 程序, 然后执行, 其结果是

```
#include <stdio.h>
int main(){
    char x = 0x82;
    printf("x = %d\n", x);
    return 0;
}
```

- A.x = -125    B.x = 130    C.x = -126    D.x = 83

2-13 将操作数 D 算术右移 k 位的指令应为( )

- A. SAL k, D
- B. SAR k, D
- C. SHR k, D
- D. SHL k, D

2-14 执行下列 C 代码后, 变量 c 和 d 的值为

```
unsigned a= 0xffffffff;  
unsigned b=2;  
int c = a;  
int d = b+c;  
printf("%d, %d\n", c, d);
```

- A.  $2^{64}-1$  和  $2^{64}+1$
- B.  $2^{64}-1$  和 1
- C. -1 和  $2^{64}+1$
- D. -1 和 1

2-15 正确指出下列 3 条指令对错的是

```
movb $-23, (%ebx)  
movq (%rbp), (%rbx)  
cmpq (%rcx, %rdi), %rax
```

- A. 错、错、对
- B. 对、对、错
- C. 错、对、对
- D. 错、对、错

2-16 下列寄存器中, 不属于被调用者保存寄存器的是( )。

- A. %rax
- B. %rbp
- C. %rbx
- D. %r12

2-17 正确指出下列 3 条指令对错的是

```
subw %ax, (%ebp)  
addl %rbx, (%rcx)  
testq (%rsi, %rcx, %rdi), %rax
```

- A. 错、错、错
- B. 错、对、对
- C. 错、对、错
- D. 对、对、错

2-18 对于某 8 位计算机, 整数用补码表示, 假设  $x=60$ ,  $y=-30$ , 则  $x+y$  的机器数及其相应的溢出标志 OF (设 1 表示溢出, 0 表示不溢出) 为 ( )。

- A. 0x1E, 0
- B. 0x1E, 1
- C. 0xE2, 0
- D. 0xE2, 1

2-19 考虑以下 C 语言代码：

```
short si=-8196;
```

```
int i=si;
```

执行上述程序段后，i 的机器数表示为

- A. 0xFFFF DFFC      B. 0x0000 9FFC  
C. 0xFFFF 9FFC      D. 0x0000 DFFC

2-20

在 Linux 下，把可执行程序 foo 反汇编生成类似于汇编代码格式的文本文件的命令是

- A. objdump -d foo foo.s  
B. objdump -d foo > foo.s  
C. disas foo > foo.s  
D. disas foo foo.s

4-1 C 语言函数 fabc 的部分代码如左下表所示，用 GCC 编译之生成汇编代码如右下表所示，分析汇编代码，完成 (1) ~ (5) 填空。

C 代码

汇编代码

```
long fabc(long *array, long n) {
    int i;
    for (i=0; i<n; i++)
    {
        if ((1) _____)
        {
            (2) _____;
        } else
            (3) _____;
        if ((4) _____)
            return (5) _____;
    }
    return -1;
}
```

```
fabc:
    movl    $0, %ecx
    jmp     .L2
.L6:
    movslq  %ecx, %rdx
    leaq    (%rdi,%rdx,8), %r8
    movq    (%r8), %rdx
    movq    %rsi, %r9
    subq    %rax, %r9
    movq    (%rdi,%r9,8), %r9
    cmpq    %r9, %rdx
    jne     .L3
    movq    %r9, (%r8)
    jmp     .L4
.L3:
    addq    %rax, %rdx
    movq    %rdx, (%r8)
.L4:
    movq    -8(%rdi,%rsi,8), %rdx
    cmpq    %rdx, (%r8)
    jg      .L5
    addl    $1, %ecx
.L2:
    movslq  %ecx, %rax
    cmpq    %rsi, %rax
    jl      .L6
    movq    $-1, %rax
.L5:
    rep ret
```

4-2

若某机器仅为 6 位，有符号数（int）用补码表示，依据以下代码填写表格

```
int z;
int x = -11;
unsigned y = x;
```

表达式	十进制表示	二进制表示
零	0	000000
z		100101
x	-11	
y		
Tmax (补码最大值)		

Answers（自行对答案得到，可能出现错误）：

判断

1 FFFFFFF

选择

1 ADACD 6 ADBDB 11 BCBDA 16 AABAB

大题

4-1

```
1 array[i] != array[n-i]
2 array[i] += i
3 array[i] = array[n-i]
4 array[i] > array[n-1]
5 i
```

4-2

表达式	十进制表示	二进制表示
零	0	000000
z	-27	100101
x	-11	110101
y	53	110101
Tmax (补码最大值)	31	011111