

1. 请写出单精度规格化小数中，负的绝对值最小值的二进制位模式，写出数值 1 的位模式。

答：负的绝对值最小值：1 00000000 000000000000000000000000

1 的位模式：0 01111111 000000000000000000000000

2. 完成以下两个单精度浮点数的求和运算 0 01111111 000000000000000000000000
000+0 01111100 011000000000000000000000。以及 0 01111111 0000000000
0000000001111+0 01111111 0110000000000000000010101

答：1) 0 01111111 001011000000000000000000

2) 0 01111111 01100000000000000000100100

3. 对于数组 `int B[8][5]`, 需要将 `B[i][j]` 保存到 `eax` 中，数组起始地址在 `rdi`，`i` 保存在 `rsi`，`j` 保存在 `rdx` 中，请完成以下代码中的空缺

```
leaq    (%rdx,%rsi, 5),%rax
```

```
leaq    (%rdi,%rax, 4),%rax
```

```
movl    (%rax, , ),%eax
```

注：答案不唯一，合理即可。

4. 已知 `int P[M][N]` 和 `int Q[N][M]`

有以下函数 `int addfun(int i,int j){ return P[i][j]+Q[j][i];}`

对应有汇编代码如下，请问 `M` `N` 各自是多少？

`addfun:`

```
movl    %edi,%edx
```

```
shll    $2,%edx
```

```
addl    %esi,%edx
```

```
movl    %esi,%eax
```

```
shll    $2,%eax
```

```
addl    %eax,%edi
```

```
movl    Q(%rdi,4),%eax
```

```
addl    P(%rdx,4), %eax
```

```
ret
```

答：M=4； N=4

5.

```
union a1{  
  
struct {  int * b1; char c1; long d1; } str1;  
  
double data[3];  
  
}
```

在 64 位系统中，按照默认的对齐方式，上述 a1 占用多少字节空间？

答：24 字节。

6. 观察以下 C 代码和对应的 x86-64 汇编代码，其中 H 和 J 是由#define 声明的常量

```
int array1[H][J];  
int array2[J][H];  
void copy_array(int x, int y) {  
    array2[y][x] = array1[x][y];  
}  
-----  
# On entry:  
# %edi = x  
# %esi = y  
#  
copy_array:  
    movslq    %edi,%rdi  
    movslq    %esi,%rsi  
    movq      %rsi, %rdx  
    salq      $4, %rdx  
    subq      %rsi, %rdx  
    addq      %rdi, %rdx  
    leaq      0(%rdi,8), %rax  
    subq      %rdi, %rax  
    addq      %rsi, %rax  
    movl      array1(%rax,4), %eax  
    movl      %eax, array2(%rdx,4)  
    ret
```

H 和 J 的值是多少？

答：H=15，J=7

7. 观察以下 x86-64 汇编代码

```
foo: # rdi = t, rsi = v
    pushq   %r12
    pushq   %rbp
    pushq   %rbx
.LCFI2:
    movq    %rdi, %rbx
    movq    %rsi, %r12
    testq   %rdi, %rdi
    je      .L3
    movl    (%rsi), %ebp
    cmpl    24(%rdi), %ebp
    jne     .L12
    jmp     .L5
.L7:
    cmpl    %ebp, 24(%rbx)
    jne     .L12
.L5:
    leal    1(%rbp), %edx
    movq    16(%rbx), %rax
    addl    (%rax,%rdx,4), %ebp
    movl    %ebp, %eax
    jmp     .L8
.L12:
    movq    %r12, %rsi
    movq    (%rbx), %rdi
    call    foo
    testl    %eax, %eax
    je      .L9
    movl    %ebp, %eax
    jmp     .L8
.L9:
    movq    8(%rbx), %rbx
    testq   %rbx, %rbx
    jne     .L7
.L3:
    movl    $0, %eax
.L8:
    popq    %rbx
    popq    %rbp
```

```
popq    %r12
ret
```

请根据汇编代码填补以下 C 代码缺失的部分：

- Node 数据结构定义如下：

```
struct Node {
    struct Node *left;
    struct Node *right;
    unsigned int *value;
    unsigned int index;
};
```

- 只使用已定义的 C 变量名，请不要直接使用寄存器名

```
int foo( struct Node* t , unsigned int * v) {
    if ( t == NULL )
        return 0;
    if( *v == t->index ) {
        return (*v + (t -> value[(*v) +1]));
    }
    return ( (foo(t->left, v)? *v : foo(t->right, v)));
}
```