1. 下列那一条指令是正确的：___c_____。

A movb $0xE, (%ebx)　B movl %rax, (%rsp)

C movq %rax, %rdx　　D movw (%rax), 4(%rsp)

2. 下列指令中不会改变条件码寄存器内容的是 ___d_____。

A　CMP　　B　TEST　　C　ADD　　D　LEA

3. 1) mov (%eax, %eax, 4), %eax

2) lea (%eax, %eax, 4), %eax

上面指令中的那一条会产生如下结果: %eax = 5 * %eax?(c)

A　1)和 2)都不会　　B　1)　　C　2)　　D　1)和 2)都会

4. 下列指令中不会改变 PC 寄存器内容的是 ___a___。

A　ADD　　B　JMP　　C　CALL　　D　RET

5. 定义结构 struct S1 {char c; int i; char d;} *p;，其所占用的内存空间为___d___个字节。

A　6　　B　7　　C　9　　D　12

6. 定义联合 typedef union {float f; unsigned u;} bit_float_t; 并执行如下函数

```
x=float bit2float(unsigned u) {
    bit_float_t arg;
    arg.u = u;
    return arg.f;
}
```

下列陈述正确的是：___C___。

A　x的值和语句(float) u得到的值相等

B　x的值和语句(float) u得到的值不相等

C　u不等于0时，x的值和语句(float) u得到的值不相等

D　以上说法都不正确

7. 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 和 N 分别是多少？

```
addfun:
    movl    %edi, %edx
    shl     $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
```

4,4

8, 已知函数 Sum_fun 的 C 语言代码及其对应的 x86-64 汇编代码框架，请补齐缺失的汇编代码

```
void Sum_fun (int *p)
{
    int a[4] = {1, 2, 4, 8}, i = 0;
    do {
        *p += a[i];
        i = i+1;
    } while(i < 4)
}
```

```
    subq    $0x10,  %rsp
    movl    $0x08,  12(%rsp)

    movl $4, 8(%rsp)        #_____
    movl    $2,     4(%rsp)
    movl    $1,     (%rsp)
    movl    $0,     %eax
    movl    $0,     %ecx
.L1
    movslq %eax, %rax

    addl (%rsp,%rax, 4), %ecx       #_____

    addq    $1, %rax            #_____

    cmpq $4, %rax               #_____
    jl      .L1
    addl    %ecx,   (%rdi)

    addq $0x10, %rsp            #_____
    ret
```

9，考虑如下一个C函数和对应的x86-64汇编代码。C代码若有缺失，请在下划线处补全，并且填写下面的跳转表。

```
int s_f (int a, int b)
{
    switch(a)
    {
        case 210:
            b *= 13;
            __break___
        case 213:
            b = 18243;
            _____________
        case 214:
            b *= b;
            __break____
        case 216:
        case 218:
            b -= a;
            __break____
        case 219:
            b += 13;
            _____________
        default:
            b -= 9;
    }
    return b;
}
```

```
40045c < s_f >:
40045c: lea     -0xd2(%rdi), %eax
400462: cmp     $0x9, %eax
400465: ja      40048a < s_f +0x2e>
400467: mov     %eax, %eax
400469: jmpq    *0x400590(, %rax, 8)
400470: lea     (%rsi,%rsi,2),%eax
400473: lea     (%rsi,%rax,4),%eax
400476: retq
400477: mov     $0x4743, %esi
40047c: mov     %esi, %eax
40047e: imul    %esi, %eax
400481: retq
400482: mov     %esi, %eax
400484: sub     %edi, %eax
400486: retq
400487: add     $0xd, %esi
40048a: lea     -0x9(%rsi), %eax
40048d: retq
```

提示：

0xd2 = 210
0x4743 = 18243

跳转表：

0x400590: _0x400470_____ 0x400598: ____0x40048a_____________
0x4005a0: _0x40048a_____ 0x4005a8: ____0x400477_____________
0x4005b0: _0x40047c_____0x4005b8: ____0x40048a_____________
0x4005c0: _0x400482_____ 0x4005c8: _____0x40048a____________
0x4005d0: _0x400482_____ 0x4005d8: ____0x400487_____________