```
    void sp_init(struct prob *sp)
    sp in %rdi
1   sp_init:
2     movl    12(%rdi), %eax
3     movl    %eax, 8(%rdi)
4     leaq    8(%rdi), %rax
5     movq    %rax, (%rdi)
6     movq    %rdi, 16(%rdi)
7     ret
```

根据这些信息，填写 sp_init 代码中缺失的表达式。

练习题 3.42　下面的代码给出了类型 ELE 的结构声明以及函数 fun 的原型：

```
struct ELE {
    long    v;
    struct ELE *p;
};

long fun(struct ELE *ptr);
```

当编译 fun 的代码时，GCC 会产生如下汇编代码：

```
    long fun(struct ELE *ptr)
    ptr in %rdi
1   fun:
2     movl    $0, %eax
3     jmp     .L2
4   .L3:
5     addq    (%rdi), %rax
6     movq    8(%rdi), %rdi
7   .L2:
8     testq   %rdi, %rdi
9     jne     .L3
10    rep; ret
```

A. 利用逆向工程技巧写出 fun 的 C 代码。

B. 描述这个结构实现的数据结构以及 fun 执行的操作。

3.9.2　联合

联合提供了一种方式，能够规避 C 语言的类型系统，允许以多种类型来引用一个对象。联合声明的语法与结构的语法一样，只不过语义相差比较大。它们是用不同的字段来引用相同的内存块。

考虑下面的声明：

```
struct S3 {
    char c;
    int i[2];
    double v;
};
union U3 {
    char c;
    int i[2];
    double v;
};
```