

vframe 对应的代码。在此使用与练习题 3.49 中同样的表示法：栈指针在第 4 行设置为值  $s_1$ ，在第 7 行设置为值  $s_2$ 。数组  $p$  的起始地址在第 9 行被设置为值  $p$ 。 $s_2$  和  $p$  之间可能有额外的空间  $e_2$ ，数组  $p$  结尾和  $s_1$  之间可能有额外的空间  $e_1$ 。

- 用数学语言解释计算  $s_2$  的逻辑。
- 用数学语言解释计算  $p$  的逻辑。
- 确定使  $e_1$  的值最小和最大的  $n$  和  $s_1$  的值。
- 这段代码为  $s_2$  和  $p$  的值保证了怎样的对齐属性？

```

1  #include <alloca.h>
2
3  long aframe(long n, long idx, long *q) {
4      long i;
5      long **p = alloca(n * sizeof(long *));
6      p[0] = &i;
7      for (i = 1; i < n; i++)
8          p[i] = q;
9      return *p[idx];
10 }
```

a) C 代码

```

long aframe(long n, long idx, long *q)
n in %rdi, idx in %rsi, q in %rdx
1  aframe:
2      pushq   %rbp
3      movq    %rsp, %rbp
4      subq    $16, %rsp           Allocate space for i (%rsp = s1)
5      leaq    30(,%rdi,8), %rax
6      andq    $-16, %rax
7      subq    %rax, %rsp           Allocate space for array p (%rsp = s2)
8      leaq    15(%rsp), %r8
9      andq    $-16, %r8           Set %r8 to &p[0]
   :
   :
   :
```

b) 部分生成的汇编代码

图 3-54 家庭作业 3.72 的代码。该函数类似于图 3-43 中的函数

- 3.73 用汇编代码写出匹配图 3-51 中函数 `find_range` 行为的函数。你的代码必须只包含一个浮点比较指令，并用条件分支指令来生成正确的结果。在  $2^{32}$  种可能的参数值上测试你的代码。网络旁注 ASM:EASM 描述了如何在 C 程序中嵌入汇编代码。
- 3.74 用汇编代码写出匹配图 3-51 中函数 `find_range` 行为的函数。你的代码必须只包含一个浮点比较指令，并用条件传送指令来生成正确的结果。你可能会想要使用指令 `cmovp`（如果设置了偶校验位传送）。在  $2^{32}$  种可能的参数值上测试你的代码。网络旁注 ASM:EASM 描述了如何在 C 程序中嵌入汇编代码。
- 3.75 ISO C99 包括了支持复数的扩展。任何浮点类型都可以用关键字 `complex` 修饰。这里有一些使用复数数据的示例函数，调用了一些关联的库函数：

```

1  #include <complex.h>
2
3  double c_imag(double complex x) {
4      return cimag(x);
5  }
6
7  double c_real(double complex x) {
8      return creal(x);
```