

```

0000000000400500 <getSum>:
 400500:      48 8b 0d 89 04 10 00    mov     1049737(%rip),%rcx
                                # 500990 <head>
 400507:      31 d2                  xor     %edx,%edx
 400509:      eb 15                  jmp     400520 <getSum+0x20>

... some nops ...

 400510:      0f be 01              movsbl (%rcx),%eax
 400513:      01 d0                  add     %edx,%eax
 400515:      0f be 51 01          movsbl 0x1(%rcx),%edx
 400519:      48 8b 49 08          mov     0x8(%rcx),%rcx
 40051d:      8d 14 10              lea    (%rax,%rdx,1),%edx
 400520:      48 85 c9              test   %rcx,%rcx
 400523:      75 eb                  jne    400510 <getSum+0x10>
 400525:      89 d0                  mov     %edx,%eax
 400527:      c3                    retq

```

What do the instructions above do? What is the mapping of registers to variables in the function `getSum`?

```

struct Node {
    char x;
    char y;
    struct Node * next;
};
struct Node * head;

int getSum(void) {
    int sum = 0;
    struct Node* p;
    for (p = head; p; p = p->next) {
        sum = sum + p->x + p->y;
    }
    return sum;
}

```

```

(gdb) p &p->x
$2 = 0x501010 "\003\001"
(gdb) p &p->y
$3 = 0x501011 "\001"
(gdb) p &p->next
$4 = (struct Node **) 0x501018
(gdb) p sizeof(struct Node)
$5 = 16

```

Why is the size of `struct Node` 16 and not 10 bytes?  
Shouldn't `&p->next = 0x501012`?