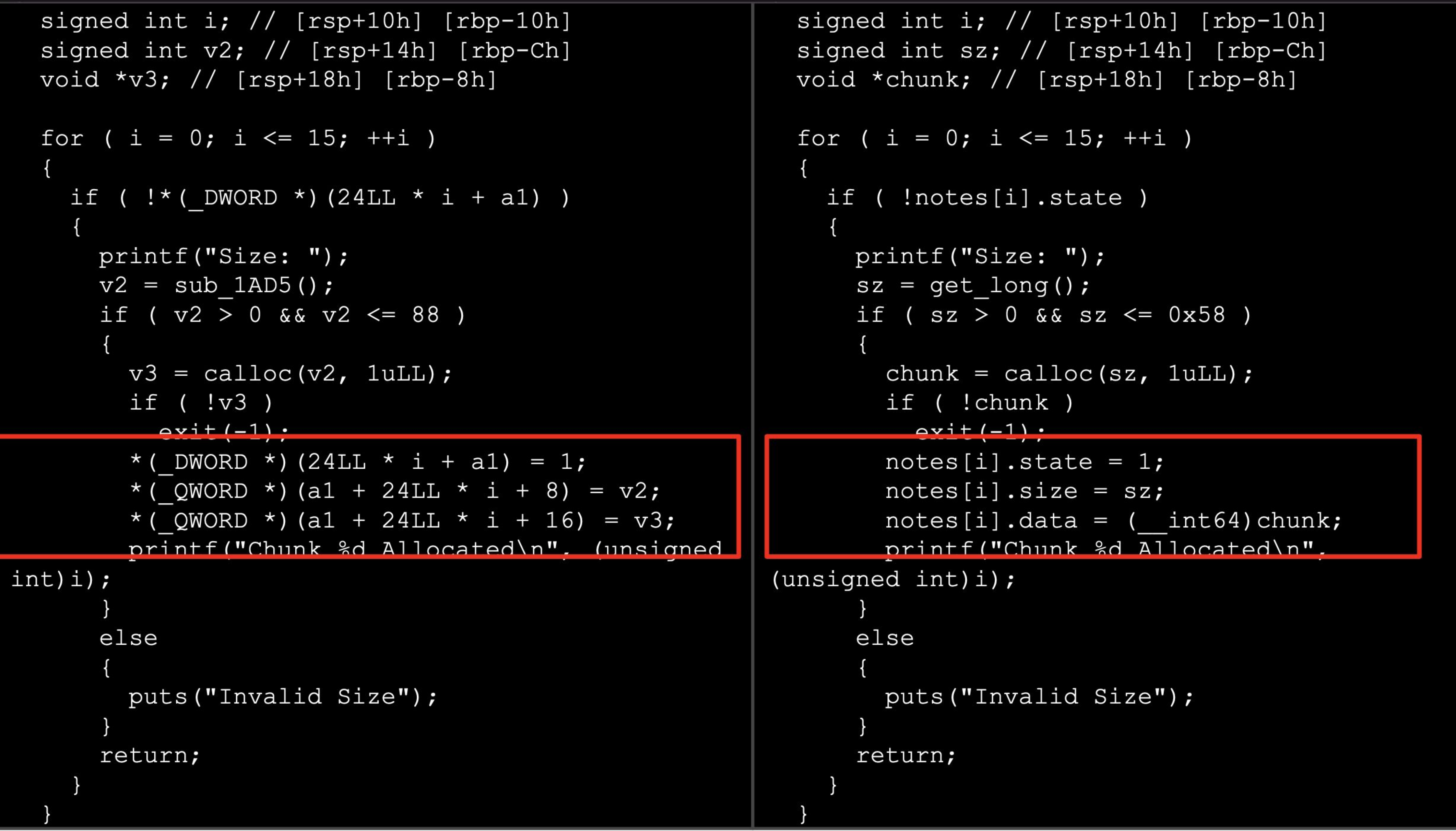

```
  signed int i; // [rsp+10h] [rbp-10h]
  signed int v2; // [rsp+14h] [rbp-Ch]
  void *v3; // [rsp+18h] [rbp-8h]

  for ( i = 0; i <= 15; ++i )
  {
    if ( !*(_DWORD *)(24LL * i + a1) )
    {
      printf("Size: ");
      v2 = sub_1AD5();
      if ( v2 > 0 && v2 <= 88 )
      {
        v3 = calloc(v2, 1uLL);
        if ( !v3 )
          exit(-1);
        *(_DWORD *)(24LL * i + a1) = 1;
        *(_QWORD *)(a1 + 24LL * i + 8) = v2;
        *(_QWORD *)(a1 + 24LL * i + 16) = v3;
        printf("Chunk %d Allocated\n", (unsigned
int)i);
      }
      else
      {
        puts("Invalid Size");
      }
      return;
    }
  }
```

```
  signed int i; // [rsp+10h] [rbp-10h]
  signed int sz; // [rsp+14h] [rbp-Ch]
  void *chunk; // [rsp+18h] [rbp-8h]

  for ( i = 0; i <= 15; ++i )
  {
    if ( !notes[i].state )
    {
      printf("Size: ");
      sz = get_long();
      if ( sz > 0 && sz <= 0x58 )
      {
        chunk = calloc(sz, 1uLL);
        if ( !chunk )
          exit(-1);
        notes[i].state = 1;
        notes[i].size = sz;
        notes[i].data = (__int64)chunk;
        printf("Chunk %d Allocated\n",
(unsigned int)i);
      }
      else
      {
        puts("Invalid Size");
      }
      return;
    }
  }
```