You are charged with maintaining a large C program, and you come across the following code:

```
typedef struct {
  int left;
  a_struct a[CNT];
  int right;
} b_struct;

void test(long int i, b_struct *bp)
{
  int n = bp->left + bp->right;
  a_struct *ap = &bp->a[i];
  ap->x[3] = n;
}
```

Unfortunately, the '.h' file defining the compile-time constant CNT and the structure a_struct are in files for which you do not have access privileges. Fortunately, you have access to a '.o' version of code, which you are able to disassemble with the dis program. Here is the disassembly of the procedure test:

```
  0x0: a0510000 ldl r2, 0(r17)
  0x4: a03100e0 ldl r1, 224(r17)
  0x8: 40410002 addl r2, r1, r2
  0xc: 42100570 s4subq r16, r16, r16
  0x10: 42011650 s8addq r16, 0x8, r16
  0x14: 42300411 addq r17, r16, r17
  0x18: b0510014 stl r2, 20(r17)
  0x1c: 6bfa8001 ret r31, (r26), 1
```

Using your reverse engineering skills, deduce the following:

A. The size of the structure a_struct.

B. The alignment requirements for a_struct.

C. The value of CNT.

D. A diagram of the complete layout for a_struct. [You may not know the names of all of the fields for this structure, but you can determine their most of the information about their types.] Mark off the different fields and indicate as much information as you can. Clearly indicate the end of the structure by crossing off any unused bytes. The diagram allows a maximum of 36 bytes.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```