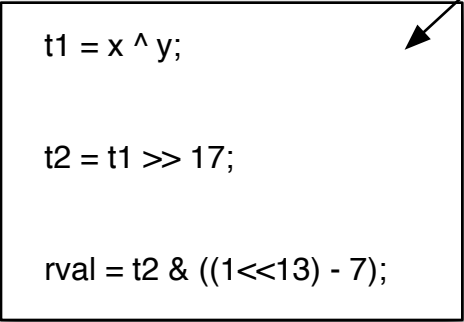


```
int x, y;  
int t1, t2;  
int rval;
```

```
int main(int argc, char *argv[])  
{  
    if (argc != 3) { fprintf(stderr, "usage: logical <num> <num>\n"); exit(1); }  
    x = atoi(argv[1]);  
    y = atoi(argv[2]);
```

Rearrange the following instruction sequence to work correctly for the C code on the left:



```
t1 = x ^ y;  
  
t2 = t1 >> 17;  
  
rval = t2 & ((1<<13) - 7);
```

```
1: mov    %eax, 0x80497b8  
2: xor    0x80497bc, %eax  
3: mov    %eax, 0x80497c0  
4: and    $0x1ff9, %eax  
5: mov    %eax, 0x80497c4  
6: mov    %eax, 0x80497c8  
7: sar    $0x11, %eax
```

```
printf("%d\n", rval);  
return 0;  
}
```

HINTS

```
y    is in 0x80497c4  
x    is in 0x80497bc  
t1   is in 0x80497c8  
t2   is in 0x80497b8  
rval is in 0x80497c0
```