

```

11. else
12.     printf("x,y,z in ascending order is %d %d %d", z, y, x);
13. }
14. }
15. else
16.     if (z < x)
17.         printf("x,y,z in ascending order is %d %d %d", z, y, x);
18.     printf("x,y,z in ascending order is %d %d %d", z, y, x);
19. else
20.     if (z < y)
21.         printf("x,y,z in ascending order is %d %d %d", x, y, z);
22.     printf("x,y,z in ascending order is %d %d %d", x, z, y);
23. else
24.     printf("x,y,z in ascending order is %d %d %d", x, z, y);
25. }
26. }
27. }

```

**Claim:** The function `sort_three()` sorts the values of  $x, y, z$  in ascending order correctly for any arbitrary  $x, y, z$ .

**Proof:** ~~Since~~ Since there are 3 variables, we can have ~~at most~~ exactly  $3! = 6$  orderings/permutations. Depending on the values, one of the permutations will be in ascending order, and this can be easily proved by doing a case by case analysis as shown in the function `sort_three()`.  $\square$

6) Given  $n$ , a positive integer, determine if  $n$  is the sum of its divisors, that is, if  $n$  is the sum of all  $t$  such that  $1 < t < n$  and  $t$  divides  $n$ . (Perfect no.)

• The full code is on my P.C.

```

1. bool check_perf(int n)
2. {
3.     int i, sum = 0;
4.     for (i = 1; i < n; i++)
5.     {
6.         if (n % i == 0)
7.             sum = sum + i;
8.     }
9.     if (sum == n)
10.        return true;
11.    else return false;

```

**Loop invariant:** The variable `sum` stores the sum of all the divisors of  $n$  from 1 to  $i-1$ .

**Initialization:** Before the 1st iteration of the loop,  $i$  is 1. The range 1 to 0 doesn't make sense, so the loop invariant trivially holds. `sum` is 0 initially.

Proof

Bo

In

In

By