Maintenance: Let the loop invariant hold for $i=K$, where $K<n-1$.
$\therefore$ sum stores the sum of all divisors of n from 1 to $K-1$.
If K divides n, condn. in line 6 holds, we add K to prev sum in line 7.
Now, $i=K+1$. sum stores the sum of all divisors of n from 1 to $(K+1)-1=K$.
If $K \nmid n$, even then, the loop variable i gets updated to $(K+1)$. sum stores the sum of all divisors of n from 1 to $(K+1)-1=K$.

$\boxed{\therefore \text{ Loop invariant holds}}$

Termination: The loop terminates when $i=n$. $\therefore$ sum stores the sum of all divisors of n from 1 to $(n-1)$. If sum$=n$, we return true indicating n is perfect, else n is not perfect and we return false □

7) The factorial function $n!$ has value 1 when $n\leq 1$ and value $n*(n-1)!$ when $n>1$. Write both a recursive an iterative C function to compute $n!$

```
1. long recursive_factorial(int n)
2. {
3.   if(n==0)
4.       return 1;
5.   else return n* recursive_factorial(n-1);
6. }
```

Claim: The recursive_factorial function correctly computes $n!$.

Proof: The proof is by mathematical induction on $K=n$, the arbitrary integer.

Base Case: $K=0$. $\because n=0$, we correctly return 1 in line 4. We all know that $0!=1$

Induction hypothesis: The recursive_factorial(K) function correctly computes $K!$, for $K=n$.

Induction Step: Now, let $K=n+1$. $\because K\neq 0$, we evaluate else block in line 5. We return $(n+1)*$ recursive_factorial$((n+1)-1=n)$

By inductive hypothesis, recursive_factorial$(n)=n!$
$\therefore (n+1)*n!=(n+1)!$ $\therefore$ For $K=n+1$, the function correctly computes $K!$ □