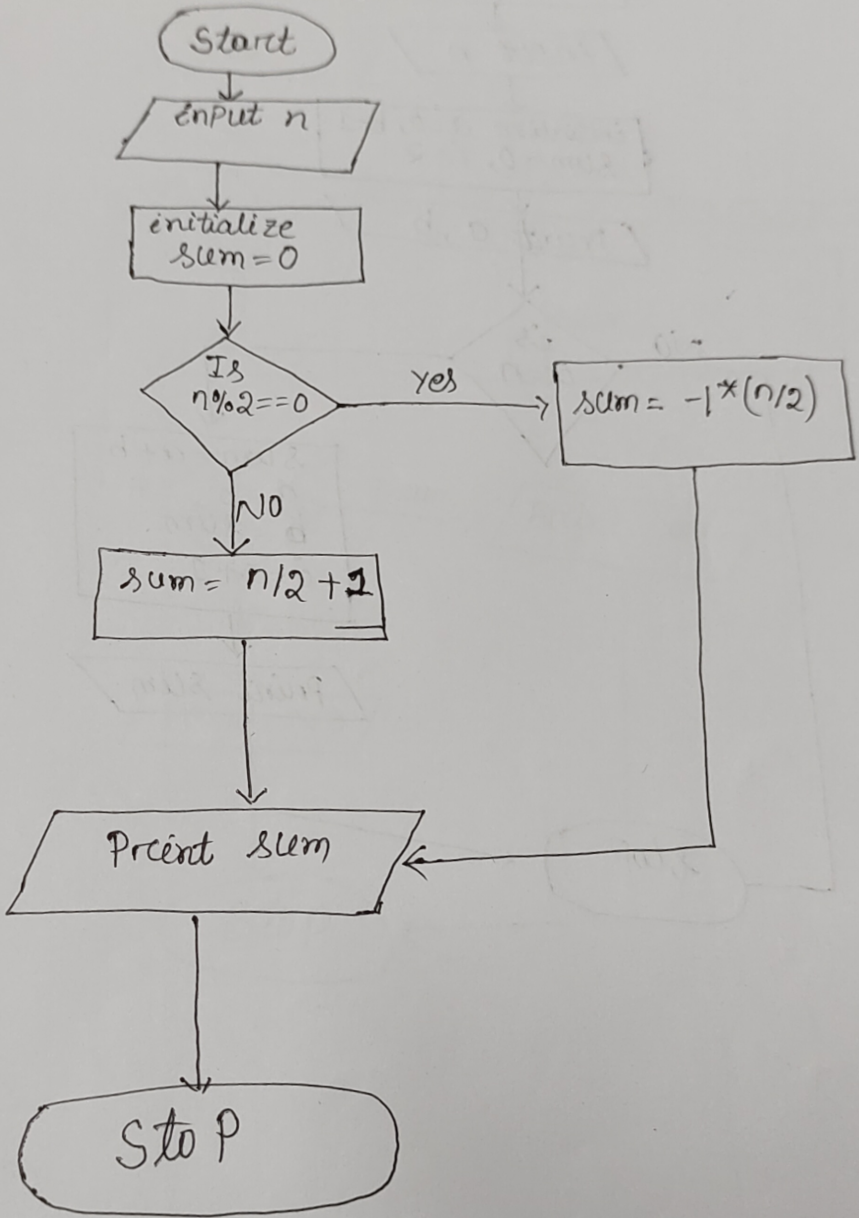
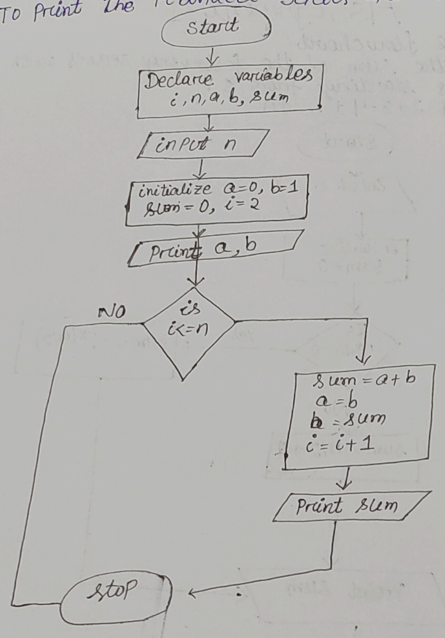
Assignment-1

1. Draw the flowchart:

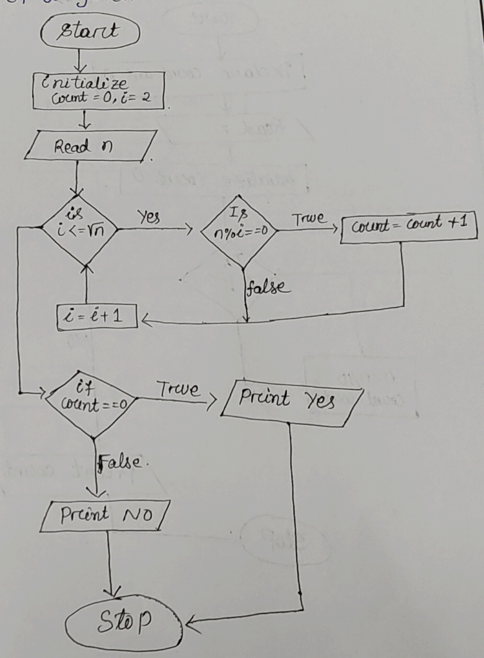
a) To find the sum of the following series with n numbers starting from 1:
sum = 1-2+3-4 ± .... ±n



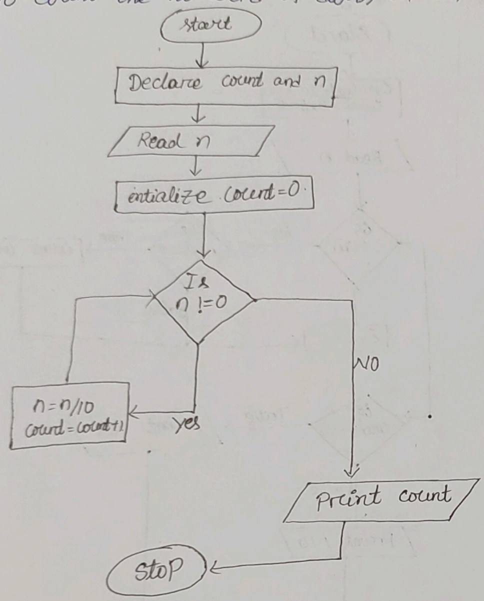
b) To Print the Fibonacci Series upto nth term



c) To check if a given number is Prime or not

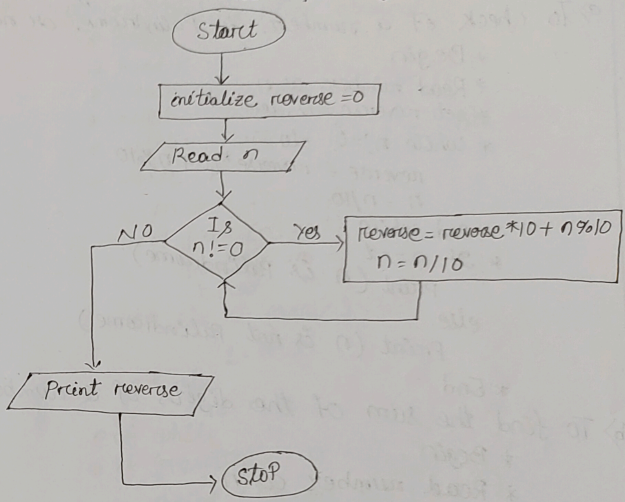


d) To count the number of digits in an integer



Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

e) To find the reverse of a given number



② write Pseudocode for the following algorithms

a) To check if a number is Palindrome or not

→ * Begin
* Read number as n
* set reverse = 0, temp = n
* while n != 0 do
    reverse = reverse * 10 + n % 10
    n = n/10
  end while
* If n == t
    Print (n is Palindrome)
  else
    Print (n is not Palindrome)
* End.

b) To find the sum of the digits of a number
* Begin
* Read number as n.
* initialize sum = 0
* while n != 0 do
    sum = sum * + n % 10
    n = n//10
  end while
* Print sum
* End

c) Iterative Binary Search on an array of n number and a given element.

→ * Begin
* Read n
* Create an array of size n
* initialize the array
* initialize Position = -1, l = 0, r = n-1, m = 0.
* while (l <= r) do
    m = l + (r - l)/2
    ** if (arr[m] == x)
        Position = m
        break
    ** else if (arr[m] < x)
        l = m+1
    ** else
        r = m-1
  end while
* if (Position == -1)
    Print (Element not Present)
  else
    Print (Element is Present at : Position)

d) To convert a given decimal number to the corresponding binary number.

* Begin
* Read n
* create an integer array binary[] of size 40
* initialize index = 0.
* while n != 0 do
    binary [index++] = n%2
    n = n/2
* for i = index-1 to 0 do
    Print (binary[i])
    i = i-1
* End.

e) To check if a number can be represented as the sum of two Prime numbers or not.

* Begin
* Read n
* intialize boolean Possible = false
* for i = 2 to n/2 do
    if IsPrime (n-i) && IsPrime(i)
    Possible = true
    break
~~* if (Possible = true)~~
* Print (possible)

* Create a function boolean IsPrime (int x)

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

* for i = 2 to $\sqrt{x}$ do
    if (x%i == 0)
    return false
  i = i+1
  end for
  return true
* End.

3) For each of the following Pairs of functions, determine whether $f(n) = O(g(n))$ or $g(n) = O(f(n))$

(a) $f(n) = n(n-1)/2$ and $g(n) = 6n$.

let us assume :-

$f(n) = O(g(n))$

$\Rightarrow f(n) \leq c.g(n)$

$\Rightarrow \frac{n(n-1)}{2} \leq c(6)(n)$

$\Rightarrow \frac{(n-1)}{12} \leq c$

$\Rightarrow \frac{n}{12} - \frac{1}{12} \leq c$

$\Rightarrow n/12 \leq c$

$\Rightarrow \boxed{c \geq n/12}$

when $n \to \infty$, then there doesn't exists any no. c which is greater than ∞.

So, $\frac{n(n-1)}{2} \leq c(6)(n)$

$\Rightarrow f(n) \leq c.g(n)$ (not possible)

Assume , $g(n) = O(f(n))$

$\Rightarrow g(n) \le c\, f(n)$

$\Rightarrow [(6n)] \le c \cdot \frac{n(n-1)}{2}$

$\Rightarrow \frac{12}{n-1} \le c$

$\Rightarrow \boxed{c \ge \frac{12}{n-1}}$

when $n \to \infty$, then $c \ge 0$, which is possible. Hence $c\, \frac{(n)(n-1)}{2} \ge 6n$

$\Rightarrow 6n \le c \cdot \frac{(n)(n-1)}{2}$

$\Rightarrow g(n) = O(f(n))$ (Proved)

b) $f(n) = n + 2\sqrt{n}$ & $g(n) = n^2$

Assume that $f(n) = O(g(n))$

$\Rightarrow f(n) \le c.\, g(n)$

$\Rightarrow n + 2\sqrt{n} \le c.\, n^2$

$\Rightarrow \boxed{1/n + \frac{2}{n^{3/2}} \le c}$

If $n \to \infty$, then $c \ge 0$ which is possible.

Hence $f(n) = O(g(n)) \to$ Proved

Now, Assume $g(n) = O(f(n))$

$\Rightarrow g(n) \le c.\, f(n)$

$\Rightarrow n^2 \le c.(n + 2\sqrt{n})$.

$\Rightarrow c \ge \frac{1}{1/n + 2/n^{3/2}}$

when $n \to \infty$ then $c \ge \infty$, which is not possible.

Hence, $g(n) \ne O(f(n))$

c) $f(n) = n + \log n$ and $g(n) = n\sqrt{n}$

$f(n) = O(g(n))$

$\Rightarrow n + \log n \le c.\, n\sqrt{n}$

$\Rightarrow n/n + \frac{\log n}{n} \le \sqrt{n}.c$

$\Rightarrow 1 + \frac{\log n}{n} \le \sqrt{n}.c$.

As when $n \to \infty$, $\frac{\log n}{n} \to 0$

So, $1 \le c\sqrt{n} \to$ which is possible

Hence, $\boxed{f(n) = O(g(n))}$

Assume $g(n) = O(f(n))$

$\Rightarrow n\sqrt{n} \le c(n + \log n)$

$\Rightarrow \sqrt{n}/c \le 1 + \log n/n$.

$\Rightarrow \frac{\sqrt{n}}{c} \le 1$ which is not possible.

so $g(n) \ne O(f(n))$.

d) $f(n) = n\log n$ and $g(n) = n\sqrt{n}/2$.

Let Assume :- $f(n) = O(g(n))$

$\Rightarrow f(n) \le c.\, g(n)$

$\Rightarrow n\log n \le c.n\, \sqrt{n}/2$.

$\Rightarrow \log n \le c\sqrt{n}$

$\Rightarrow c \ge \frac{\log n}{\sqrt{n}}$, when $n \to \infty$, then $\frac{\log n}{\sqrt{n}} \to 0$

$\Rightarrow c \ge 0 \to$ which is possible.

Hence, $f(n) \le c.\, g(n)$

$\Rightarrow \boxed{f(n) = O(g(n))}$

Assume :- $g(n) = O(f(n))$

$\Rightarrow g(n) \leq C.f(n)$

$\Rightarrow \frac{n\sqrt{n}}{2} \leq C.\ n\log n.$

$\Rightarrow C \geq \frac{\sqrt{n}}{2\log n}$, so when $n \to \infty$, then $\frac{\sqrt{n}}{2\log n} \to \infty$

$\Rightarrow C \geq \infty \to$ which is impossible.

Hence $g(n) \neq O(f(n))$.

(c) $f(n) = 2(\log n)^2$ and $g(n) = \log n + 1$

Assume that :- $f(n) = O(g(n))$

$\Rightarrow 2(\log n)^2 = O(\log n + 1)$

$\Rightarrow 2(\log n)^2 \leq C.\log n + 1.$

$\Rightarrow C \geq \frac{(\log n)^2}{\log n + 1}$ & when $n \to \infty$, then $C \geq \infty$

So, $f(n) \neq O(g(n))$

Assume that : $g(n) = O(f(n))$

$\Rightarrow \log n + 1 \leq C.2(\log n)^2$

$\Rightarrow \frac{\log n + 1}{2(\log n)^2} \leq C$

$\Rightarrow \frac{1}{2\log n} + \frac{1}{2(\log n)^2} \leq C.$

$\Rightarrow$ when $n \to \infty$, then $\log n$ & $(\log n)^2 \to \infty$

$\Rightarrow 0 \leq C$. (Possible).

So, $\boxed{g(n) = O(f(n))}$ Proved

(4) State TRUE OR FALSE justifying your answer with proper reason.

(a) $2n^2 + 1 = O(n^2)$

$2n^2 + 1 \leq 2n^2 + n^2$

$\Rightarrow 2n^2 + 1 \leq 3n^2$

So, $C = 3$ & $f(n) = n^2$

So, $\forall n \geq$, we have $T(n) = O(n^2)$ (True)

(b) $n^2(1 + \sqrt{n}) = O(n^2)$

$n^2(1 + \sqrt{n}) = n^2 + n^2\sqrt{n}$

we have :- $n^2 + n^2\sqrt{n} \leq n^2\sqrt{n} + n^2\sqrt{n}$

$\Rightarrow n^2 + n^2\sqrt{n} \leq 2n^2\sqrt{n}$

$C = 2$, $f(n) = n^2\sqrt{n}$

Hence for $\forall n \geq 1$, we have $T(n) \leq 2n^2\sqrt{n}$

$\Rightarrow T(n) = O(n^2\sqrt{n}) \neq O(n^2)$.

So (false)

(c) $n^2(1 + \sqrt{n}) = O(n^2\log n)$.

$n^2(1 + \sqrt{n}) = n^2 + n^2\sqrt{n}$

$\Rightarrow n^2 + n^2\sqrt{n} \leq n^2\sqrt{n} + n^2\sqrt{n}$

$\Rightarrow n^2 + n^2\sqrt{n} \leq 2n^2\sqrt{n}$

$C = 2$, $f(n = n^2\sqrt{n})$

Hence for $\forall n \geq 1$, we have $T(n) \leq 2n^2\sqrt{n}$

$\Rightarrow T(n) = O(n^2\sqrt{n}) \neq O(n^2\log n)$

(false)

(d) $3n^2+\sqrt{n} = O(n+n\sqrt{n}+\sqrt{n})$

$3n^2+\sqrt{n} \le 3n^2+n^2$

$\Rightarrow 3n^2+\sqrt{n} \le 4n^2$

$c=4$ & $f(n)=n^2$

Hence for $\forall n \ge 1$, we have $T(n) \le 4n^2$

$\Rightarrow T(n) = O(n^2)$

we have been given $O(n+n\sqrt{n}+\sqrt{n})$

$= O(n\sqrt{n})$

Now, $T(n) = O(n.n)$.

$\ne O(n\sqrt{n})$.

As $n^2 (n.n) \ge (n\sqrt{n})$

so, (False)

(e) $\sqrt{n}\log n = O(n)$

we know that Growth rate of $\log n \le \sqrt{n}$

$\Rightarrow \sqrt{n}\log n \le \sqrt{n}.\sqrt{n}$

$\Rightarrow \sqrt{n}\log n \le n$.

$\Rightarrow c=1$ & $f(n)=n$.

Hence $\forall n \ge 1$, we have $T(n) \le n$

$\Rightarrow T(n) = O(n)$.

(True)

(f) $\lg n \in O(n)$.

we know.

$1 < \lg n < \sqrt{n} < n < n\log n < n^2 < n^3 < n! \ldots$

so $\lg n = O(\lg n)$

$= O(n)$ As $\lg n \le n$.

So (True)

(g) $n \in O(n\lg n)$

we know $1 \le \log n$

$\Rightarrow 1(n) \le (n)\log n$

$\Rightarrow$ So $c=1$ $f(n)=n\log n$.

Hence $\forall n \ge 1$, we have $T(n) \le n\log n$

$\Rightarrow T(n) = O(n.\lg n)$

$\therefore$ (True)

(h) $n\lg n \in O(n^2)$

$\lg n \le n$.

$\Rightarrow n\lg n \le n^2$

So, $c=1$, $f(n)=n^2$.

Hence $\forall n \ge 1$, we have $T(n) \le n^2$

$\Rightarrow T(n) = O(n^2)$

$\therefore$ (True)

i) $2^n \in \Omega(6^{\ln n})$

we have :- $6^{\ln n}$

$= n^{\ln 6}$

$= n^{1.79}$

So, from the series of Growth rate of func

we have :- $n^{1.79} \le 2^n$

$\Rightarrow 2^n \ge n^{1.79}$.

$c=1$, $f(n)=n^{1.79}$

So $\forall n \ge 1$, we have :- $T(n) \ge n^{1.79}$

$\Rightarrow T(n) = \Omega(n^{1.79})$

$= \cancel{\Omega\, 6^{\ln n}}$

$= \Omega\, 6^{\ln n}$

$\therefore$ (True)

j) $lg^3 n \in O(n^{0.5})$

we know $\log n \le \sqrt{n}$

$\Rightarrow (lg\, n)(lg\, n)(lg\, n) \le (\sqrt{n})(\sqrt{n})(\sqrt{n})$

$\Rightarrow lg^3 n < n^{3/2}$

$\Rightarrow lg^3 n < n^{1/2}$

$c = 1$ & $f(n) = n^{1/2}$.

Hence $\forall n \geq 1$ we have $T(n) < n^{1/2}$

$\Rightarrow T(n) = O(n^{1/2})$.

$\therefore$ (True)

5) For each of the following pair of functions $f(n)$ and $g(n)$, determine whether $f(n) = O(g(n))$ or $f(n) = \Theta(g(n))$ or $f(n) = \Omega(g(n))$

(a) $f(n) = \sqrt{n}$, $g(n) = \log(n+3)$

we know $\sqrt{n} \geq \log n$, $\forall n \geq 1$.

$\Rightarrow \sqrt{n} \geq (1)(\log n + 3)$

$\Rightarrow c = 1$ & $f(n) = \log(n+3)$.

So, $\forall n \geq 1$, we have $\boxed{f(n) = \Omega\, g(n)}$

(b) $f(n) = n\sqrt{n}$, $g(n) = n^2 - n$

$f(n) = \Theta(n\sqrt{n}) = O(n\sqrt{n})$

$g(n) = O(n^2)$

$\sqrt{n} \le n$

$\Rightarrow n\sqrt{n} \le n^2$.

$\Rightarrow n\sqrt{n} \le c_1(n^2 - n)$

$\Rightarrow f(n) \le c.\, g(n)$. $\Rightarrow \boxed{f(n) = O(g(n))}$ Ans

(c) $f(n) = 2^n - n^2$, $g(n) = n^4 + n^2$

$T(n)$ for $f(n) = O(2^n)$

$T(n)$ for $g(n) = O(n^4)$.

So from the growth of rate of function

$2^n \geq c.n^4$

$\Rightarrow f(n) \geq c.g(n)$

$\Rightarrow \boxed{f(n) = \Omega(g(n))}$ Ans

(d) $f(n) = n^2 + 3n + 4$, $g(n) = 6n^2$

$T(n)$ for $f(n) = O(n^2)$

$T(n)$ for $g(n) = O(n^2)$

$n^2 \le cn^2$

$\Rightarrow n^2 = n^2$

$\boxed{f(n) = \Theta(g(n))}$ Ans

(e) $f(n) = n + n\sqrt{n}$, $g(n) = 4n\log(n^2+1)$

$T(n)$ for $f(n) = O(n\sqrt{n})$

~~T(n) for g(n) = 4n log (n²+1)~~

$T(n)$ for $g(n) = O(n\log n)$

From the series of growth rate of func.

we have $\log n \le \sqrt{n}$

$\Rightarrow \sqrt{n} \geq \log n$

$\Rightarrow n\sqrt{n} \geq n\log n$

$\Rightarrow f(n) \geq c.g(n)$

So, $f(n) = \Omega\, g(n)$.

6) Prove the following statements :-

a) If $f_1(n) = \Omega(g_1(n))$ and $f_2(n) = \Omega(g_2(n))$ then $f_1(n)+f_2(n) = \Omega(g_1(n) + g_2(n))$

We have, $f_1(n) = \Omega(g_1(n))$

$f_2(n) = \Omega(g_2(n))$

$0 \le c_1 g_1(n) \le f_1(n)$ —(I)

$0 \le c_2 g_2(n) \le f_2(n)$ —(II)

Adding (I) & (II) we get

$0 \le c_1 g_1(n) + c_2 g_2(n) \le f_1(n) + f_2(n)$

$\Rightarrow 0 \le (g_1(n)+g_2(n)) \le f_1(n)+f_2(n)$

$\Rightarrow f_1(n)+f_2(n) = \Omega(g_1(n)+g_2(n))$. (proved)

b) If $f_1(n) = \Omega(g_1(n))$ and $f_2(n) = \Omega(g_2(n))$ then $f_1(n)+f_2(n) = \Omega(\min(g_1(n)+g_2(n))$

Given $f_1(n) = \Omega(g_1(n))$

$f_2(n) = \Omega(g_2(n))$

$0 \le c_1 g_1(n) \le f_1(n)$ —(I)

$0 \le c_2 g_2(n) \le f_2(n)$ —(II)

Adding (I) & (II) we get :-

$0 \le c_1 g_1(n) + c_2 g_2(n) \le f_1(n) + f_2(n)$

$\Rightarrow 0 \le c_3(g_1(n)+g_2(n)) \le f_1(n)+f_2(n)$

$\Rightarrow 0 \le c(g_1(n)+g_2(n)) \le f_1(n)+f_2(n)$

Here c is the minimum value s.t. it is less than $f_1(n)+f_2(n)$

$\rightarrow$ It is happening as $c_3 > c$, but still less than $f_1(n)+f_2(n)$

So, $c(g_1(n)+g_2(n)) \le f_1(n)+f_2(n)$

Hence, $f_1(n)+f_2(n) = \Omega(\min(g_1(n)+g_2(n)))$

(proved)

c) If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$ then $f_1(n).f_2(n) = O(g_1(n).g_2(n))$.

$\rightarrow$ Given $f_1(n) = O(g_1(n))$

$\Rightarrow 0 \le f_1(n) \le c_1(g_1(n))$ —(I)

$f_2(n) = O(g_2(n))$

$\Rightarrow 0 \le f_2(n) \le c_2(g_2(n))$ —(II)

Multiplying (I) & (II), we get :-

$0 \le f_1(n).f_2(n) \le c_1c_2(g_1(n).g_2(n))$

$\Rightarrow 0 \le f_1(n).f_2(n) \le c_3(g_1(n).g_2(n))$

Hence $f_1(n).f_2(n) = O(g_1(n).g_2(n))$ (proved)

d) If $f(n) = O(g(n))$, then $f(n)^k = O(g(n)^k)$

Given :- $f(n) = O(g(n))$

$\Rightarrow 0 \le f(n) \le c_1(g(n))$

Multiplying the power k to inequalities, we get

$0^k \le f(n)^k \le c_1^k g(n)^k$

$\Rightarrow 0 \le f(n)^k \le c_2(g(n)^k)$

So, $f(n)^k = O(g(n)^k)$

(Proved)