Assignment No. 1

① check if given no. is odd or Even

→ 1. start

2. Enter Number

3. If (No. /2 == 0)

4. If condition true No. is even

5. If condition false No. is odd.

② Factrial of Number using recursion

→ 1. start

2. Enter Number

3. If (Number > 1)

4. condition true →

4. let Result = 1

5. If cond^n true While (No. == 1)
     { No. × Result = Result

6. Result No. decrease by 1 }

7. Display result

8. end.

3. factorial of No. using Recursion

→ 1. start

2. Enter No.

3. Give pass No. to function

int Fact (n)

    { if (n ==1)

       return 1;

       else

       return n * Fact(n-1);

   }

4. end.

4. swap two numbers without using 3rd variable

→ 1. start

2. Enter num1, num2

3. num1 = num1 + num2

4. num2 = num1 - num2

5. num1 = num1 - num2

6. Display num1 & num2

7. end.

5. Check given no. is positive or negative.

→ 1. Start

2. Enter Number

3. If (Num >= 0)

4. Condition true → Number is positive

5. Condition false → Number is Negative

6. End.

6. Given Number is leap year or NOT

→ 1. Start

2. Enter year

3. Check year / 400

4. If condition true display given year is leap year

5. If cond^n false (year / 100)

6. If condition2 true display given year is leap year

7. If cond^2 false (year / 4)

8. If condition3 true display given year is leap year

9. If condition3 false display given
   year is not leap year.
10. end

7. Print No. 1 to 10 without using
   loop
→ 1. start
   2. Create array of size 10 & store value.
   3. Display value of each index
   4. end.

8. Print the digits of given Number.
→ 1. start
   2. Enter Number & digit While (Number >
   3. i=(Number % 10)
   4 display i
   5. Number = Number/10

9. Print all the factors of given Number

→ 1. start

2. Enter Number

3. for int $i = 1$ to $i <= n/2$ & increment i by 1

4. if $(n \% i == 0)$

5. if condition true display i

—10. 6. End.

10. sum of digits of given Number.

→ 1. start

2. Enter Number   3. sum = 0

4. While (Number > 0)

) {   { int i = Number $\% 10$

~~int~~ sum = i + sum

Number = Number / 10

}

5. display sum.

11. Find smallest number of a,b,c

→ 1. start

2. Enter Number a,b,c

3. if a ≤ b ≠ and a ≤ c

4. display a

5. else compare (b ≥ c)
   smaller
6. If b is ~~greatest~~ than c display
   b else cond^n true

7. ~~ets 7.~~ else cond^n false display c.

12. Add Numbers without using arithmatic operator.

→ 1. start

2. Enter Num1 & Num2

3. while (Num2 != 0)

4. Carry is equal to Num1 bitwise AND Num2

5. Num1 = Num2 bitwise XOR Num1

6. Num2 = carry << 1

7. display Num1.   8. end.

13. Reverse given number

→ 1. start

2. Enter number

3. Number While (Number > 0)

4. i = Number % 10

5. display i

6. Number = Number / 10.

7. end.

14. Find GCD of Two given number.

→ 1. start

2. Number num1, num2, GCD = 1 & i

2. Enter num1, num2

3. GCD = 1 & i

4. for i = 1 to i <= num1 && i <= num
   & increase i by 1

5. if (num1 % i == 0) && (num2 % i ==
6. display gcd = i

7. end.

15. LCM of two given Number

→ 1. Start

2. Enter Num1, Num2

3. Lcm = (n1 > n2) ? n1 : n2

4. while (true)

5. if (Lcm % num1 == 0 & Lcm %.
num2 == 0) then break
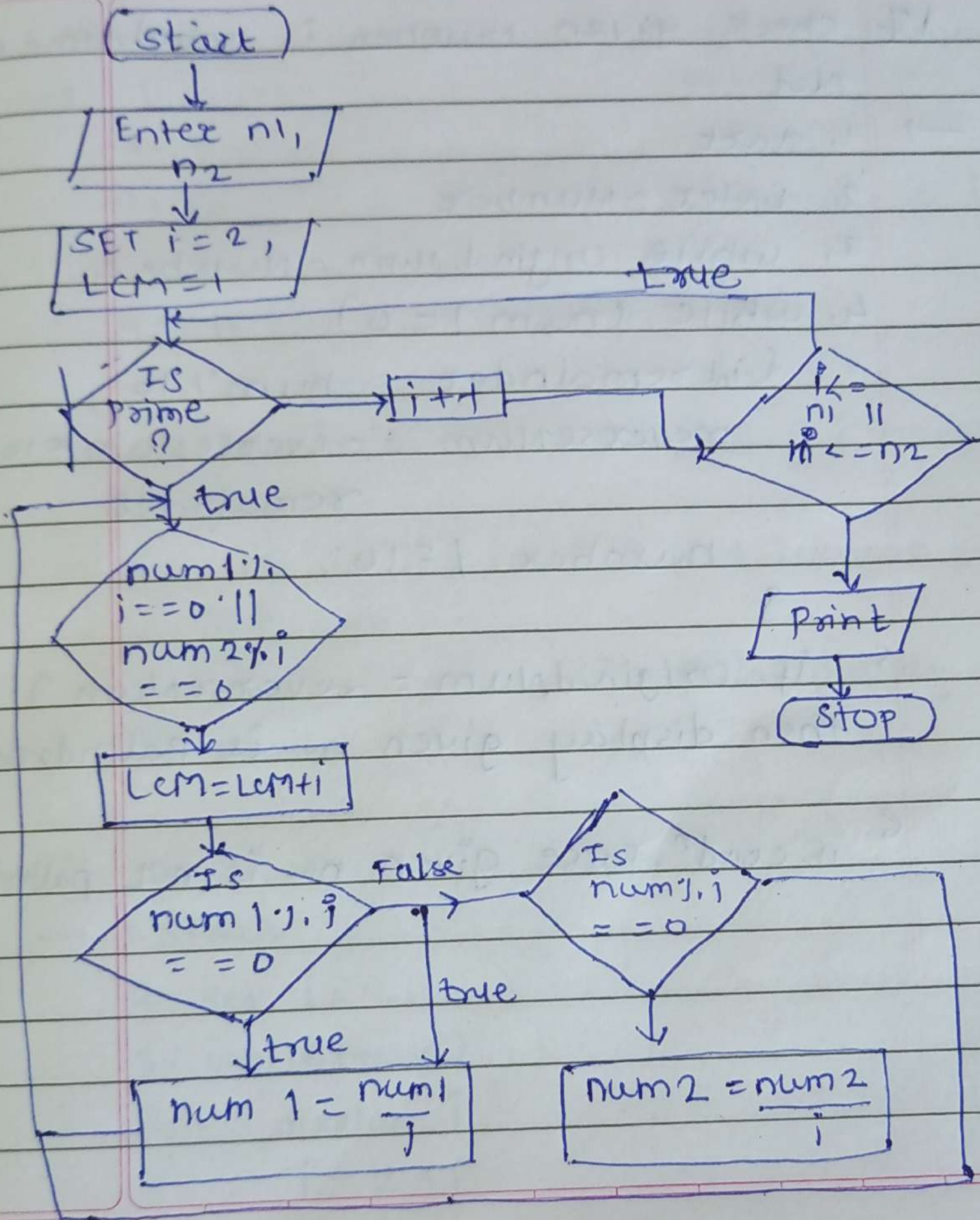display Lcm & break.

6. if not then increment Lcm by 1

7. end.

16. Lcm of two numbers using the prime factor method.

→ 1. start

2. Enter num1 & num2

3. q1 & q2 etc and find gcd

```
( Start )
    │
    ▼
/ Enter n1, /
/    n2    /
    │
    ▼
/ SET i = 2, /
/  Lcm = 1  /
    │
    ▼
  < IS          >───────→ [ i++ ]
  < Prime ?     >
    │ true
    ▼
  < num1 %      >
  < i == 0 ||   >
  < num2 % i    >
  <   == 0      >
    │
    ▼
[ Lcm = Lcm+i ]
    │
    ▼
  < IS          >──── False ───→  < IS       >
  < num1 % j    >                 < num % i  >
  <  == 0       >                 <  == 0    >
    │ true              │ true          │
    ▼                   ▼               ▼
[ num1 = num1/j ]  [ ... ]     [ num2 = num2/i ]
```

true

< i <=
  n1 ||
  i <= n2 >

/ Print /
    │
    ▼
(Stop)

17 Check given Number is palindrome or Not.

→ 1. start

2. Enter Number

3. while original_Num = Number

4. while (num != 0)

   { int remainder = num % 10;

      reverseNum = reverseNum * 10 + remainder;

      number /= 10;

   }

5. if (originalNum = reverseNum) then display given no. is palindrome

6. if cond^n false given. no. is not palindrome

18. Print all prime factors of Numbers

→ 1. start

2. Enter number

3. for i = 2 to i <= number/2 & i increase by 1

4. if (num % i == 0)

$$isprime = 1;$$

5. for j = 2 to j <= i/2 & j increase by 1

6. if (i % j == 0) set isprime = 0; & break

7. if (isprime == 1) then display i.

8. end.

19. print even Number series.

→ 1. start

2. set i = 2

3. while (true)

4. display i

5. i = 2 + i

6. end.

20. Print odd Number
→ 1. Start
2. set i = 1
3. while true
4. display i
5. i = 2 + i ;
6. end.