19.09

1. WAP to accept a number and check whether the number is odd or even without using %, /, and any if construct.

2. WAP to find the quotient and remainder of a division without using %, /

3. WAP to print 1st 10 odd fibonacci terms

4. WAP to print each term and sum of the following series. 5 + 8 + 12 + 17 + ---- + upto n terms.

1.
```
if (num % 2 == 0)
    S.O.P("even");
else
    S.O.P("odd");
```

2.
```
void main(int a, int b)
{
  int quo = 0;
  for (; a >= b; a -= b)
      quo++;
  S.O.P("quotient" + quo);
  S.O.P("Remainder" + a);
}
```

```
void main(int num)
{
  int i;
  for (i = num; i >= 2; i -= 2)
      ;
  S.O.println(i == 0 ? "Even" : "odd");
}
```

3. 1, 1, ~~2~~, 3, 5, ~~8~~, 13, 21, ~~34~~, 55

```
void main ( )
{  int i, a = 1, b = 0, c ;
   for (i = 1 ; i <= 10 ; )
     { c = a + b ;
       if (c % 2 == 1)
       { s.o printf(c) ;
         i++ ;
       }
       a = b ;
       b = c ;  } }
```

i	a	b	c	s.o(c)
1	1	0	1	1
2	0	1	1	1
3	1	1	2	—
3	1	2	3	3
4				

4. 5 + 8 + 12 + 17 + - - -

```
void main (int n)
{
  int diff=3, num=5, i, sm=0;
  for(i=1; i<=n; i++)
  {
    S.o print(num);
    num= num + diff;
    diff = diff + 1;
  }
}
```

```
sum= sm + num;
S.o P(num + " ");
S.o println("sum of the series" + sm);
```

Write a program to accept a sum of money and calculate the different number of denominations of Rs. 100, 50, 10, 2, 1 notes..

Example : Amount = 568

Output : 100 x 5 = 500

50 x 1 = 50

10 x 1 = 10

2 x 4 = 8

```
void main(int amt)
{
    int n100, n50, n10, n2, n1 ;
n100 = amt/100 ;
amt = amt % 100 ;
n50 = amt/50 ;
amt = amt % 50 ;
n10 = amt/10 ;
amt = amt % 10 ;
n2 = amt/2 ;
amt = amt % 2 ;
n1 = amt ;
```

```
if (n100 > 0)
s.o.p("100 x " + n100 + " = " + (n100*100));
if (n50 > 0)
s.o.p("50 x " + n50 + " = " + (n50*50));
. . . .
- . . -
}
```

26.09 1) WAP to accept a number and check whether the number belongs to fibonacci series or not. Example: Input 6 output: Not in fib series
Input 8 output: Found in fib series.

2) WAP to accept a number and display the prime decomposition of the number. Example
Input 8 output - 2, 2, 2
Input 30 output - 2, 3, 5

3) WAP to display the highest and lowest factors excluding 1 and the number itself.
INPUT 10 output - 5, — highest
2 - lowest.

4) WAP to display the multiplication table of a number x upto y times.
=

Solution.

1)

```
void main(int n)
{
  int i, a=1, b=0, c ;
 for(i=1 ; c<=n ; i++)
 { c= a+b;   → 1, 1, 2, 3, 5, 8
   if(c == n) ↵
   { S.o.p("Found in fib seris"); ↵
     return; ↵
   }
   a = b;
   b = c;
 }
 S.o.p("Not found in fib seris");
}
```

n = 6 / n = 8

2) Prime decomposition.

num = 8 → 2, 2, 2

num = 30 → 2, 3, 5

```
Void main (int num)
{
  int i  j
 for (i= 2; i <= num ; i++)
  if (num % i == 0)
   { S.o.printf(" i + " ");   2 3 5
  num /= i ;  →   15/3 = 5
  i--
 }
 else i++ ;
}
```

9.

```
void main ( int num)
{
  int i ;
  for ( i= 2; i < num ; i++)
  {
    if (num % i == 0 )
    {
      S.o pln ( " Lowest factor = " + i ) ;
      S.o pln( " Highest factor = " + (num / i) ) ;
      return ; / break ;
    }
  }
```

4. void main (int x, int y)
{ int i ;
for (i = 1; i<=3; i++)
S.o.pln(x + "X" + i + "=" + (x * i));
}

x = 6
y = 10

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18

03.10 Jump Statement.

i) break → It is used in Loop or switch block to take the control out of the block.

ii) continue → It is used in loop structure to skip the part of the loop beyond it and proceed for next iteration.

iii) return → It is used to stop the execution of a program and to pass the control to the calling function.

```
for (i = 1; i <= 8; i++)
{   if (i % 7 == 0)
        break;
    s.o.p(i);
}
```

```
for (i = 1; i <= 8; i++)
{   if (i % 3 == 0)
        continue;
    s.o.print(i);
}
```

break can be used in switch or Loop only
continue " " " in Loop only
} NOT with if only

return can be used any where in a program within or outside loop.

output:

2 3 4 4 5 5 6 7 ~~8~~ 8

```
for(i=2; i<=7; s.o.print(++i))
{
    if(i%3==0)
        continue;
    s.o.print(i);
}
```

1). WAP to accept a number and check the number is triangular number or not.

Example:
3 → 1+2
6 → 1+2+3
10 → 1+2+3+4

If a number is the sum of all the natural number from 1, then it is triangular number.

2) WAP to display all the triangular number upto a given number

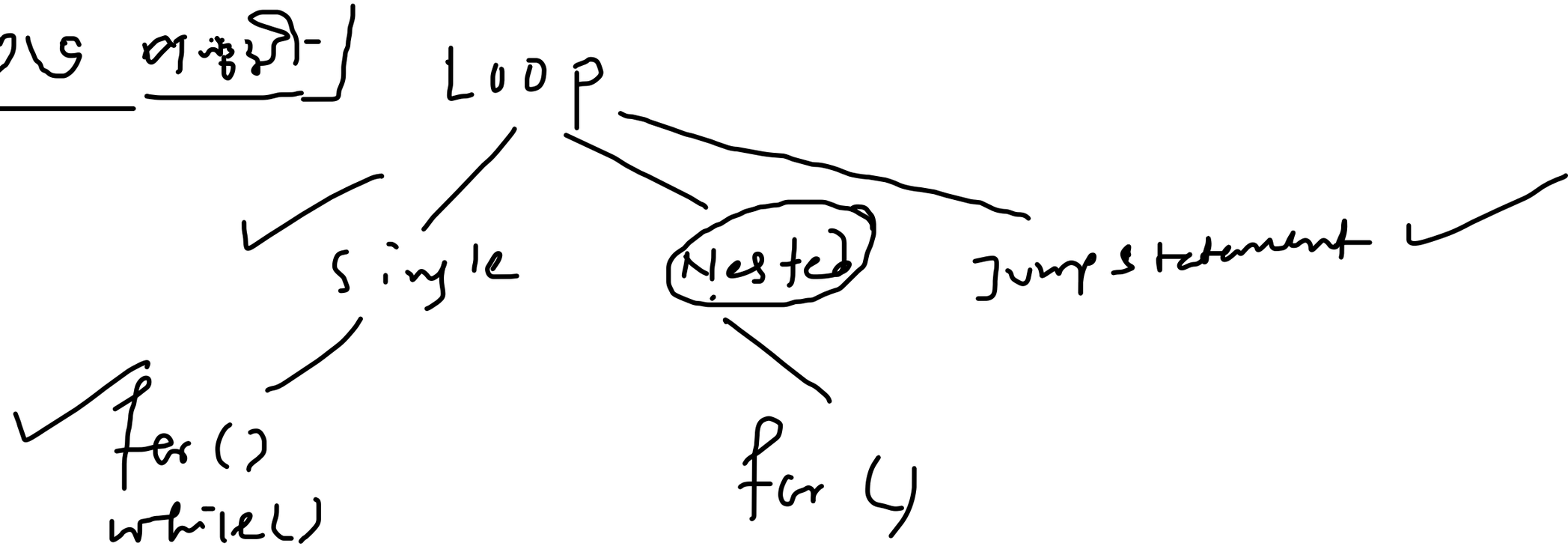
3) WAP to print 1st n triangular number.

1)
```
void main (int num)
{
    int i, sum=0;
    for (i=1; i<=num; i++)
    {
        sum=sum+i;
        if (sum==num)
        {
            S.o.p ln ("Triangular");
            return;
        }
    }
    S.o.p ln ("Not Triangular");
}
```

2)
```
void main (int num)
{
 int i ; sm=0;
 for (i=1; sum<num; i++)
  {
    sum=sum+i;
    S.o.p(sum+" ");
  }
}
```

3)
```
void main (int n)
{
 int i, sm=0;
 for (i=1; i<=n; i++)
  {
    sm=sm+i;
    S.o.p(sm+" ");
  }
}
```

10.10.21 তারিখ পাঠশালা



Nested Loop : A loop within another loop is called nested loop.

```
for (i = 1; i <= 5; i += 3)
{
    for (j = i; j <= 7; j += 4)
        System.out.print(j);
    System.out.print(i);
}
```

Find the output ↑

i	j	print
1	1	1
	5	5
4		9
	4	1
		4
		8
		4

1. WAP to print each term and sum of the following series.

$$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \text{- - - n terms.}$$

2. WAP to print 1st n prime numbers

3. WAP to print 1st n terms of fibonacci Series in reverse order.

1. void main (int x, int n)

```
{  int j,i, k = 1, f ;
   double term , sum = 0 ;
for ( i = 1 ; i <= n ; i++ )
{   f = 1;
    for (j = 1; j <= k ; j++)
        f = f * j;
   term = Math.pow (x, k) / f ;
S.O.P (term + " ");
sum += term ; k += 2;
}
S.O.P(sum);
}
```

sum += term ;
sum = sum + term

→ 1, 2, 3

$x^k / f \rightarrow x^1/1$

$x^3 / f \rightarrow x^3/6$

n = 4

```
void main ( int n)
{  int i, j, num = 2 ;
  for (i = 1 ; i <= n ;      )
{
                int count = 0 ;
     for ( j = 1 ; j <= num ; j ++)
   { if (num % j == 0)
                  count ++ ; }
  if (count == 2)
{      S.o.p ( num ) ;
       i ++ ; }
num ++ ;
}
```

prime check

9.
```
void main (int n)
{
 int i, j, a, b, c=0;
  for( i= n;  i>= 1; i--)
{
      a= 1;
      b= 0 ;
  for (j = 1 ; j <= i; j++)
    {   c= a + b;
        a = b;
        b = c;
    }
  S.o.p ( c +" ");
}
```

n = 5

5, 3, 2, 1, 1

21.10 ಅಭ್ಯಾಸ ಕಾರ್ಯಕ್ರಮ.

Nested Loop

- for() within for()
- for() within while() / do ... while()
- while() / do while()
  - within while() / do while()
  - while() within for()

Nested Loop ⇒ pattern printing.

i) WAP to check a number Krishnamurty or Special number or not.

145 ⇒ 1! + 4! + 5!

ii) WAP to print all the prime number in a range x, y (y > x)

iii) WAP to print all the 3 digit spy number.

1124 → $1+1+2+4$ = $1\times1\times2\times4$

8 8

iv) WAP to print all the 3 digit armstrong number. $153 = 1^3+5^3+3^3$

1) Void main (int num)
{ int copy = num, d, sum = 0, fact, i;
while (num>0)
{ d = num % 10; ⟶ fact = 1;
for (i = 1; i <= d; i++)
fact = fact * i;
sum = sum + fact;
num /= 10; }

153 → 5 → 5! → sum
4 →
num

if (sum == copy)
S.o.Pln ("Special");
else S.o.Pln ("Not Special");
}

ii)
```
void main (int x, int y)
{
  int i, count, j ;
  for (i = x; i <= y; i++)
  {
    count = 0 ;
    for (j = 1; j <= i; j++)
      if (i % j == 0)
        count++ ;
    if (count == 2)
      S.o.println (i + " ") ;
  }
}
```

iii)
```
void main ( )
{ int i, d, sum = 0, prod = 1, copy ;
  for (i = 100; i <= 999; i++)
  { copy = i; sum = 0; prod = 1;
    while (copy > 0)
    { d = copy % 10;
      sum += d;
      prod *= d;
      copy /= 10;
    }
    if (sum == prod)
      S.o.p (i + " ");
  }
```

iv)

```
void main()
{   int i, sum, copy ;
for (i= 100 ; i<= 999; i++)
{   copy= i ; sum= 0 ;
   while (copy > 0)
   { d= copy % 10 ;
     sum= sum + d*d*d;
     copy = copy / 10
   }
   if (sum == i)
          S.o.p( i + " ");
}}
```

125 % 10 ⇒ 5

153

↓

$153 = 1^3 + 5^3 + 3^3$

(int) Math.pow(d, 3)