1. Factorial:-

```
READ n
INITIALIZE fact=1
FOR (i=n; i>=0; i--):
  IF i==0:
    RETURN fact
  fact = fact*i
```

2. Palindrome of Number:-

```
READ n
INITIALIZE reverse=0, num=n
WHILE num>0:
  reminder = num%10
  reverse = (reverse*10)+reminder
  num = num/10
IF n==reverse:
  PRINT "Palindrome"
ELSE:
  PRINT "Not Palindrome"
```

3. Fibonacci Series:-

```
READ n       //n-->limit
INITIALIZE f1=0, f2=1, f3=0
IF n==1:
  PRINT(f1)
ELSE:
  PRINT(f1,f2)
FOR (i=3; i<=n; i++):
  f3 = f1+f2
  f1 = f2
  f2 = f3
  PRINT(f3)
```

4. Greatest of three numbers:-

```
READ a,b,c
IF a>b:
   IF a>c:
      RETURN a
   ELIF c>b:
        RETURN c
   ELSE:
        RETURN  b
```

5. Odd or Even:-

```
READ n
IF n%2==0:
   RETURN "Even"
ELSE:
   RETURN "Odd"
```

6. Check for prime:-

```
READ n
IF n<1:
   RETURN "Number should be a Natural Number"
ELSE IF n==1:
   RETURN "Neither Prime nor Composite"
FOR (i=2; i<n; i++):
   if n%i==0:
      RETURN "Not a Prime"
RETURN "Prime Number"
```

7. Check Armstrong Number:-

```
READ n
INITIALIZE l=len(str(n)), num=n, sum=0
WHILE n>0:
   rem = n%10
   sum = sum + rem**l
```

```
    n = n/10
IF num==sum:
    RETURN True
ELSE:
    RETURN False
```

8. GCD of Two Numbers:-

```
READ n1, n2
INITIALIZE r=0
INITIALIZE a = (n1>n2) ? n1 : n2   //Greater number
INITIALIZE b = (n1>n2) ? n2 : n1   //Smaller number
WHILE (a%b != 0):
    r = a%b
    a = b
    b = r
RETURN r
```

9. LCM of Two Numbers:-

```
READ n1, n2
INITIALIZE max = (n1>n2) ? n1 : n2   //Finding max of n1 and n2
WHILE (True):
    IF (max%n1==0) && (max%n2==0):
        BREAK
    max = max+1
RETURN max
```

10. Sum of Digits:-

```
READ n
INITIALIZE sum=0
sum = (n*(n+1))/2
RETURN sum
```

11. Reverse a Number:-

```
READ num
INITIALIZE reverse=0
WHILE num>0:
  reminder = num%10
  reverse = (reverse*10)+reminder
  num = num/10
RETURN reverse
```

12. Count Vowels and Consonants:-

```
READ string
INITIALIZE vowel_count=0, cons_count=0
FOR (i=0; i<len(string); i++):
  IF string[i] IN ['a','e','i','o','u','A','E','I','O','U']:
    vowel_count += 1
  ELSE:
    cons_count += 1
RETURN vowel_count, cons_count
```