

Assignment 5 pseudocodes

1. Problem 1 solution

```
CLASS MyRectangle  
BEGIN
```

```
    METHOD MAIN  
    BEGIN
```

```
        READ user_input for width and height of rectangle  
        width ← user_input value for width  
        height ← user_input value for height
```

```
        area ← Area(width, height)  
        perimeter ← Perimeter(width, height)
```

```
        PRINT Entered width:  width  
        PRINT Entered height:  height
```

```
        IF (isValid(width, height)==true) THEN  
            PRINT Area:  area  
            PRINT Perimeter:  perimeter  
        ELSE  
            PRINT This is invalid rectangle. Try again.  
        ENDIF
```

```
    END MAIN  
    METHOD isValid(w, h)  
    BEGIN
```

```
        IF((w+h)>30) THEN
```

```
            RETURN true
```

```
        ELSE
```

```
            RETURN false
```

```

        ENDIF
    END isValid

    METHOD Area(w, h)
    BEGIN

        Area ← w*h
        RETURN Area
    END Area

    METHOD Perimeter(w, h)
    BEGIN

        Perimeter ← 2*(w+h)
        RETURN Perimeter
    END Perimeter
END MyRectangle

```

2. Problem 2 solution

```

CLASS FeetMeters
BEGIN

    METHOD MAIN
    BEGIN

        PRINT  Feet    Meter    Meter    Feet
        k ← 1.0
        WHILE (k ≤ 20.0)
            PRINT  k    feetToMeter(k)    k    meterToFeet(k)
            k ← k+1
        ENDWHILE
    END MAIN

```

```

METHOD feetToMeter(j)
BEGIN

    meter ← 0
    FOR( n ← 1.0; n ≤ j; n ← n+1)
        meter ← 0.305*n
    ENDFOR
    RETURN meter
END feetToMeter

```

```

METHOD meterToFeet(j)
BEGIN

    feet ← 0
    FOR(n ← 1.0; n ≤ j; n ← n+1)
        feet ← 3.279*n
    ENDFOR
    RETURN feet
END meterToFeet
END FeetMeters

```

3. Problem 3 solution

```

CLASS PrintTableSeries
BEGIN

    METHOD MAIN
    BEGIN

        READ user_input for an integer
        n ← user_input for an integer

        PRINT i      Sum(i)
        i ← 1
        WHILE (i ≤ n)
            sum ← displaySums(i)
            PRINT i      displaySums(i)

```

```
        i←i+1
    ENDWHILE
END MAIN
```

```
METHOD displaySums(k)
BEGIN
```

```
    sum←0
    FOR (i←1.0;i<=k;i←i+1)
```

```
        sum←sum + ((i/(i+1)))
    ENDFOR
```

```
    RETURN sum
END displaySums
```

```
END PrintTableSeries
```

4. Problem 4 solution

```
CLASS PalindromePrime
BEGIN
```

```
    METHOD MAIN
    BEGIN
```

```
        total←50
        displayperline←10
        count←1
        number←2
```

```
    WHILE (count<=total)
        IF (isPrime(number) && isPalindrome(number))THEN
            PRINT  number
```

```
        IF(count % displayperline==0)THEN
            PRINTLINE
```

```

        ENDIF
        count ← count+1
        number ← number+1
    ELSE
        number ← number+1
    ENDIF
ENDWHILE
END MAIN
METHOD isPrime(num)
BEGIN
    IF (num==2) THEN
        RETURN true
    FOR (divisor ← 2; divisor ≤ num/2; divisor ← divisor+1)
        IF (num % divisor == 0)
            RETURN false
        ENDIF
    ENDFOR
    RETURN true
END isPrime

METHOD isPalindrome(num)
BEGIN
    k ← num
    test ← 0
    WHILE (num != 0)
        lastdigit ← num % 10
        test ← test*10+lastdigit
        num ← num/10
    ENDWHILE

    RETURN (k==test)
END isPalindrome
END PalindromePrime

```

