Pseudocode	FORTRAN (original)	
INPUT number	10	FORMAT(I4)
	20	READ 10, INUM
INPUT divisor	30	READ 10, IDIVSR
intermediate <-- number	40	IMED = INUM
WHILE intermediate >= divisor	50	IF (INUM – IDIVSR) 80, 60, 60
intermediate <-- intermediate – divisor	60	IMED = IMED – IDIVSR
END WHILE	70	GOTO 5
remainder <-- intermediate	80	IREMR = IMED
OUTPUT remainder	90	PUNCH 1, IREMR

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

COBOL (probably buggy, modified from "Hello World" program found on the Internet)

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. MODULO.
000300
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100310     ACCEPT NUMBER.
100320     ACCEPT DIVISOR.
100330     MOVE NUMBER TO INTERMEDIATE.
100340     PERFORM UNTIL INTERMEDIATE < DIVISOR
100350         SUBTRACT DIVISOR FROM INTERMEDIATE
100360     END-PERFORM.
100400     MOVE INTERMEDIATE TO REMAINDER
100500     DISPLAY REMAINDER.
100600     STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800     EXIT.
```

Pseudocode	BASIC (interactive, from the original 1964 Dartmouth manual)	
INPUT number	10	DATA user_types_value_of_N
	15	READ N
INPUT divisor	20	DATA user_types_value_of_D
	25	READ D
intermediate <-- number	30	LET I = N
WHILE intermediate >= divisor	40	IF I >= D GO TO 70
intermediate <-- intermediate – divisor	50	LET I = I – D
END WHILE	60	GO TO 40
remainder <-- intermediate	70	LET R = I
OUTPUT remainder	80	PRINT R

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

C

```
#include <stdio.h>

void main(int argc, char* argv)
{
    int number, divisor, intermediate;

    scanf(“%d”, number);
    scanf(“%d”, divisor);

    intermediate = number;

    while (intermediate >= divisor)
    {
        intermediate = intermediate – divisor;
    }

    remainder = intermediate;

    printf(“%d”, remainder);
}
```

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

FORTRAN 77

```
PROGRAM MODULO
INTEGER NUMBER, DIVISOR
INTEGER INTERMEDIATE

READ *, NUMBER
READ *, DIVISOR

INTERMEDIATE = NUMBER

DO WHILE (INTERMEDIATE .GE. DIVISOR)

    INTERMEDIATE = INTERMEDIATE - DIVISOR

END DO

REMAINDER = INTERMEDIATE

PRINT *, REMAINDER
```

Pseudocode

```
INPUT number
INPUT divisor
intermediate <-- number
WHILE intermediate >= divisor
    intermediate <-- intermediate – divisor
END WHILE
remainder <-- intermediate
OUTPUT remainder
```

BASIC (structured, as used in early Personal Computers)

```
INPUT number
INPUT divisor
intermediate = number
WHILE intermediate >= divisor
    intermediate = intermediate - divisor
WEND
remainder = intermediate
PRINT remainder
```

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

Java (not object oriented, console input and output)

```
import java.io.*;

public class Modulo
{
    public static void main(String[] args)
    {
        int number, divisor, intermediate, remainder;

        number = Integer.valueOf(System.console().readLine());
        divisor = Integer.valueOf(System.console().readLine());

        intermediate = number;

        while (intermediate >= divisor)
        {
            intermediate = intermediate – divisor;
        }

        remainder = intermediate;

        System.out.println(remainder);
    }
}
```

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

Visual Basic (not object oriented, console input and output)

```
Module Modulo

    Sub Main()

        Dim number, divisor, intermediate, remainder As Int32

        Int32.TryParse(System.Console.In.ReadLine(), number)
        Int32.TryParse(System.Console.In.ReadLine(), divisor)

        intermediate = number

        While (intermediate >= divisor)

            intermediate = intermediate – divisor

        End While

        remainder = intermediate

        System.Console.Out.WriteLine(remainder)

    End Sub

End Module
```

Pseudocode

```
INPUT number

INPUT divisor

intermediate <-- number

WHILE intermediate >= divisor

    intermediate <-- intermediate – divisor

END WHILE

remainder <-- intermediate

OUTPUT remainder
```

C# (not object oriented, console input and output)

```
using System;

class Program
{
    static void main(string[] args)
    {
        int number, divisor, intermediate, remainder;

        Int32.TryParse(System.Console.In.ReadLine(), out number);
        Int32.TryParse(System.Console.In.ReadLine(), out divisor);

        intermediate = number;

        while (intermediate >= divisor)
        {
            intermediate = intermediate – divisor;
        }

        remainder = intermediate;

        System.Console.Out.WriteLine(remainder);
    }
}
```