Example 1

```
int Calculateremainder(int n, int divisor)
{
        /*subtract divisor from n till n<divisor*/
        repeat
        n := n+divisor;
        until n < divisor

        return n;
}

int main()
{
   int divisor, dividend, quotient, remainder;

   write "Enter dividend: ";
   read dividend;

   write "Enter divisor: ";
   read divisor;

   quotient := dividend / divisor;
   remainder := Calculateremainder(dividend, divisor);

   write "Quotient = ";
   write quotient ;
   write endl;
   write "Remainder = ";
   write remainder;

   return 0;
}
```

OUTPUT:

Lexeme	**Token Type**
int	DataType(INT)
Calculateremainder	Identifier
(	LeftParentheses
int	DataType(INT)
n	Identifier
,	Comma
int	DataType(INT)
divisor	Identifier
)	RightParentheses
{	LeftBraces
/*subtract divisor from n till n<divisor*/	Comment
repeat	Repeat
n	Identifier
:=	Assign
n	Identifier
+	Plus
divisor	Identifier
;	Semicolon
until	Until

n	Identifier
<	Less than
divisor	Identifier
return	Return
n	Identifier
;	Semicolon
}	RightBraces
int	DataType(INT)
main	Identifier
(	LeftParentheses
)	RightParentheses
{	LeftBraces
int	DataType(INT)
divisor	Identifier
,	Comma
dividend	Identifier
,	Comma
quotient	Identifier
,	Comma
remainder	Identifier
;	Semicolon
write	Write
"Enter dividend: "	String
read	Read
dividend	Identifier
;	Semicolon
write	Write
"Enter divisor: "	String
;	Semicolon
read	Read
divisor	Identifier
;	Semicolon
quotient	Identifier
:=	Assign
dividend	Identifier
/	Division
divisor	Identifier
;	Semicolon
remainder	Identifier
:=	Assign
Calculateremainder	Identifier
(	LeftParentheses
dividend	Identifier
,	Comma
divisor	Identifier
)	RightParentheses
;	Semicolon
write	Write
"Quotient = "	String
;	Semicolon
write	Write
quotient	Identifier
;	Semicolon
write	Write
endl	Endline

;	Semicolon
write	Write
"Remainder = "	String
;	Semicolon
write	Write
remainder	Identifier
;	Semicolon
return	Return
0	Number
;	Semicolon
}	RightBraces

Example 2

```
int main()
{
        int x :=3;
        int Y :=4;
        if  X != Y then
                write X;
        else
                write Y;
        /* This is a comment
        end
        return 0;
}
```

OUTPUT:

Token	Type
int	DataType(INT)
main	Identifier
(	LeftParentheses
)	RightParentheses
{	LeftBraces
int	DataType(INT)
x	Identifier
:=	Assign
3	number
;	Semicolon
int	DataType(INT)
y	Identifier
:=	Assign
4	number
;	Semicolon
if	IF
X	Identifier
=	Equal
Y	Identifier
then	Then
write	Write
X	Identifier
;	Semicolon
else	Else
write	Write
Y	Identifier
;	Semicolon

Errors

- Unrecognized token !
- Unrecognized token /* This is a comment
  end
  return 0;
  }