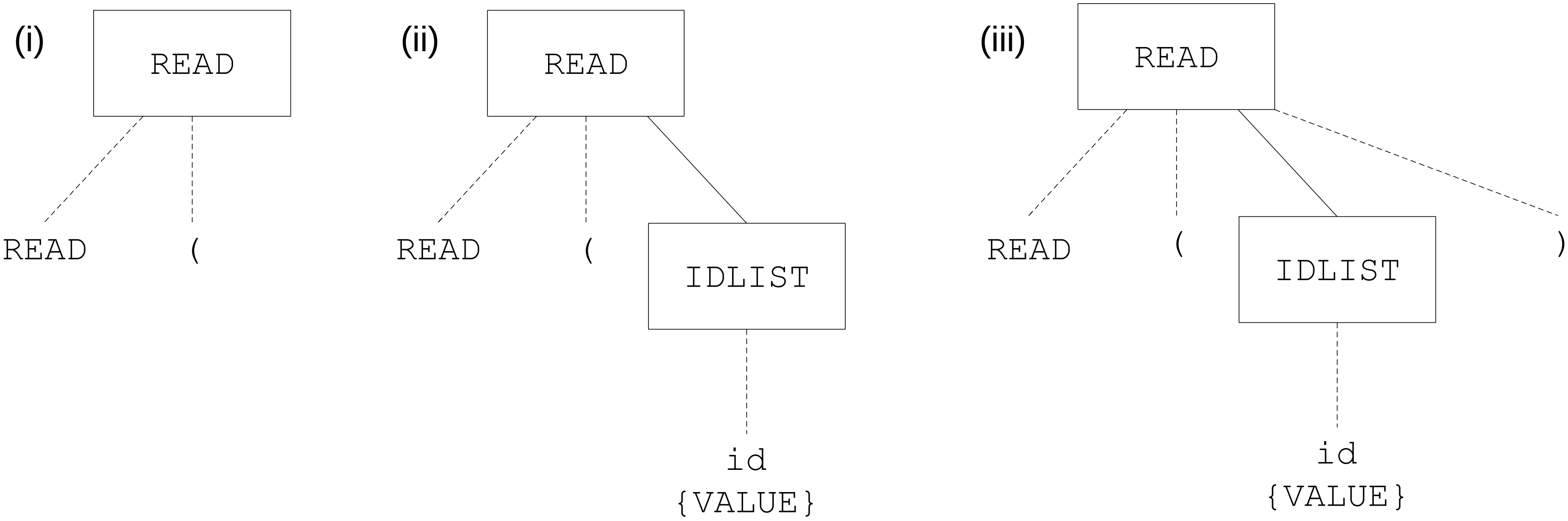
Top-Down

```
procedure READ
begin
    FOUND := FALSE
    if TOKEN = 8 {READ} then
    begin
        advance to next token
        if TOKEN = 20 {(} then
        begin
            advance to next token
            if IDLIST returns success then
                if TOKEN = 21 {)} then
                begin
                    FOUND := TRUE
                    advance to next token
                end {if )}
        end {if (}
    end {if READ}
    if FOUND = TRUE then
        return success
    else
        return failure
end {READ}
```

```
procedure IDLIST
begin
    FOUND := FASLE
    if TOKEN = 22 {id} then
    begin
        FOUND := TRUE
        advance to next token
        while (TOKEN = 14 {,}) and (FOUND = TRUE) do
        begin
            advance to next token
            if TOKEN = 22 {id} then
                advance to next token
            else
                FOUND := FALSE
            end {while}
        end {if id}
        if FOUND = TRUE then
            return success
        else
            return failure
end {IDLIST}
```


(i)
READ
READ
(
(ii)
READ
READ
(
IDLIST
id
{VALUE}
(iii)
READ
READ
(
IDLIST
)
id
{VALUE}

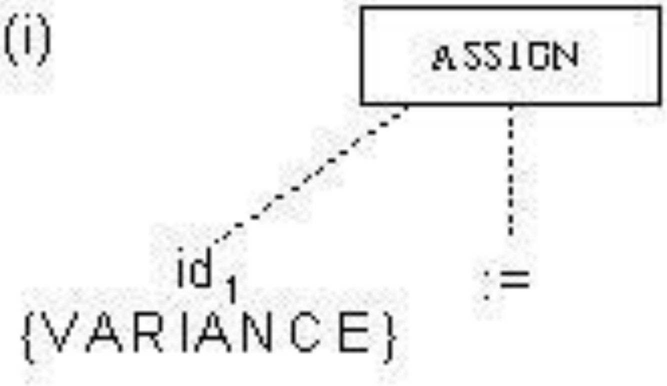
```
procedure ASSIGN
begin
    FOUND := FALSE
    if TOKEN = 22 {id} then
    begin
        advance to next token
        if TOKEN = 15 {:=} then
        begin
            advance to next token
            if EXP returns success then
                FOUND := TRUE
        end {if :=}
    end {if id}
    if FOUND = TRUE then
        return success
    else
        return failure
end {ASSIGN}
```

```
procedure EXP
begin
    FOUND := FALSE
    if TERM returns success then
    begin
        FOUND := TRUE
        while ((TOKEN = 16 {+}) or (TOKEN = 17 {-}))
        and (FOUND = TRUE) do
        begin
            advance to next token
            if TERM returns failure then
                FOUND := FALSE
        end {while}
    end {if TERM}
    if FOUND = TURE then
        return success
    else
        return failure
end {EXP}
```

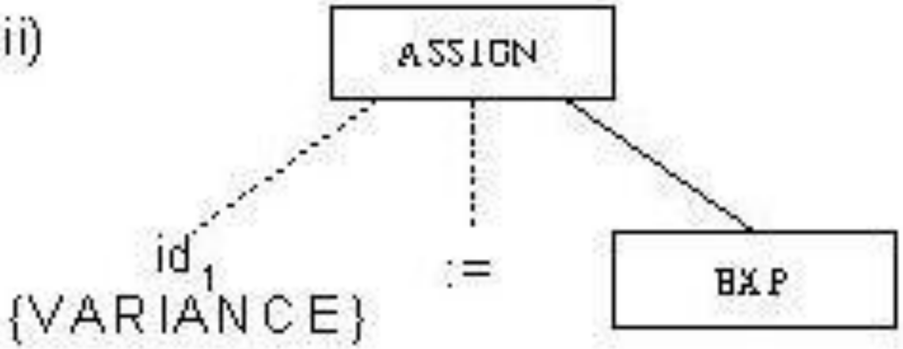
```
procedure TERM
begin
  FOUND := FALSE
  if FACTOR returns success then
  begin
     FOUND := TRUE
     while ((TOKEN = 18 {*}) or (TOKEN = 19 {DIV}))
     and (FOUND = TRUE) do
     begin
          advance to next token
          if FACTOR returns failure then
               FOUND := FALSE
     end {while}
  end {if FACTOR}
  if FOUND = TRUE then
     return success
  else
     return failure
end {TERM}
```

```
procedure FACTOR
begin
    FOUND := FALSE
    if (TOKEN = 22 {id}) or (TOKEN = 23 {int}) then
    begin
        FOUND := TRUE
        advance to next token
    end {if id or int}
    else  if TOKEN = 20 {(} then
    begin
        advance to next token
        if EXP returns success then
            if TOKEN = 21 {)} then
            begin
                FOUND := TRUE
                advance to next token
            end {if )}
    end {if (}
    if FOUND = TRUE then
        return success
    else
        return failure
end {FACTOR}
```

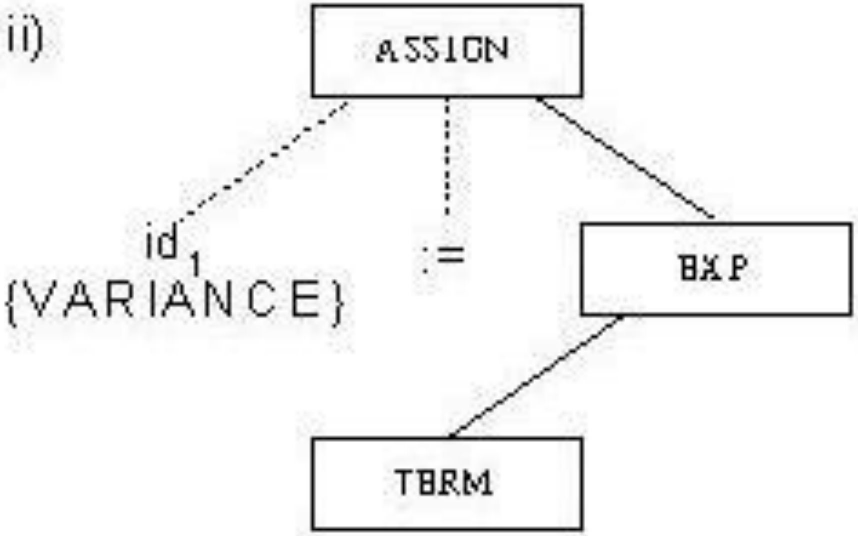
(i)

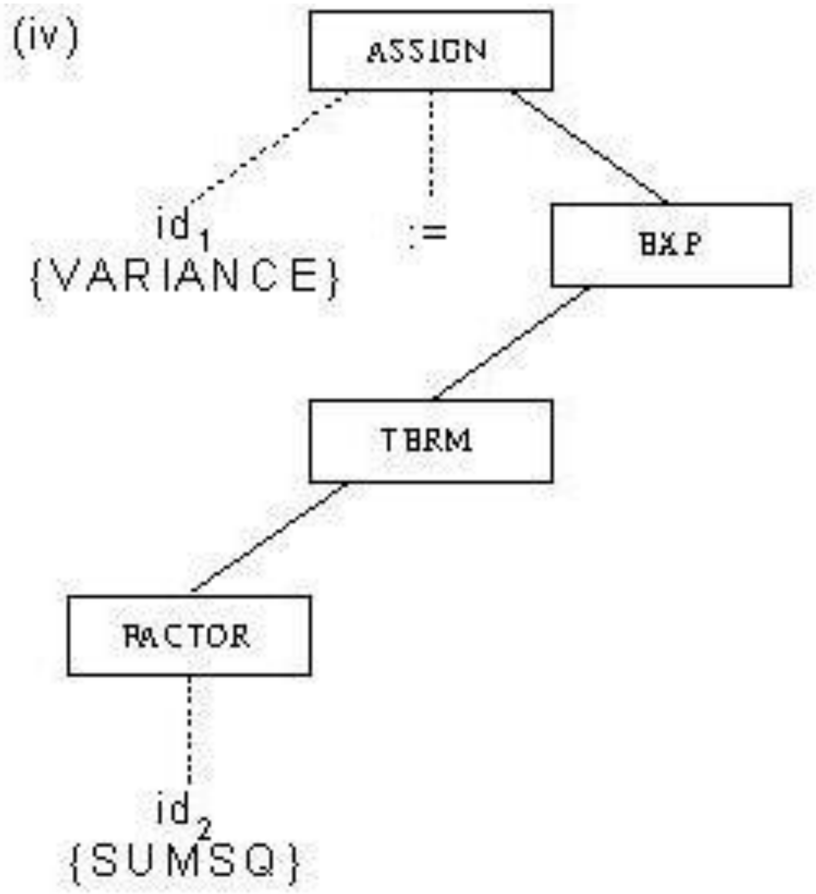

ASSIGN
id_1
{VARIANCE}
:=

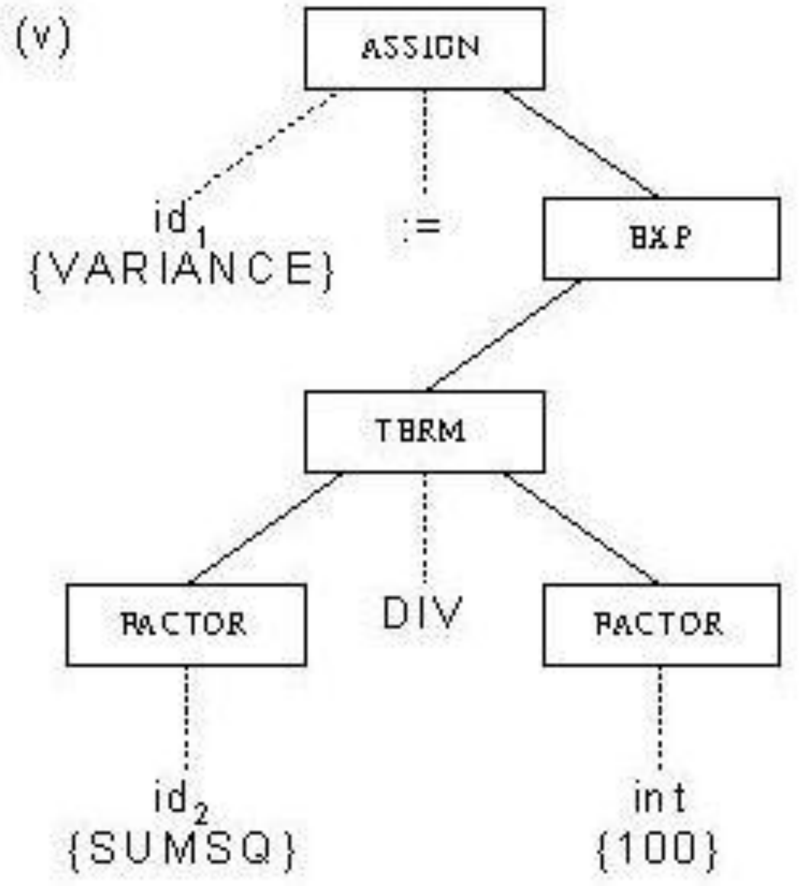
(ii)


ASSIGN
id_1
{VARIANCE}
:=
EXP

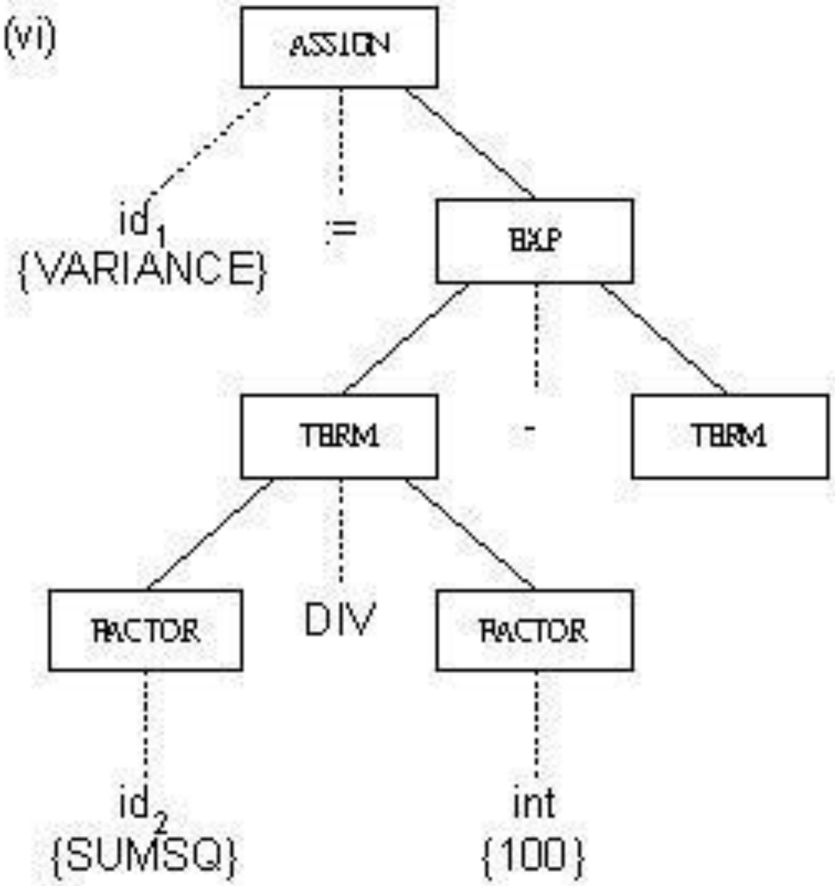
(iii)


ASSIGN
id_1
{VARIANCE}
:=
EXP
TERM

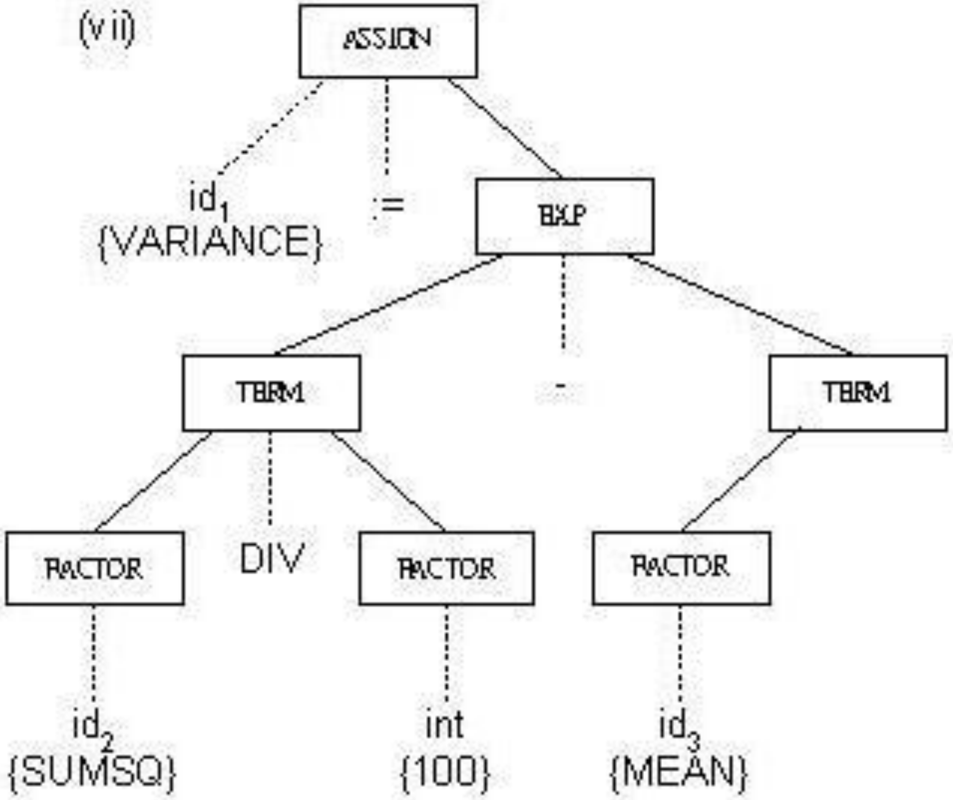

(iv)
ASSIGN
id$_1$
{VARIANCE}
:=
EXP
TERM
FACTOR
id$_2$
{SUMSQ}

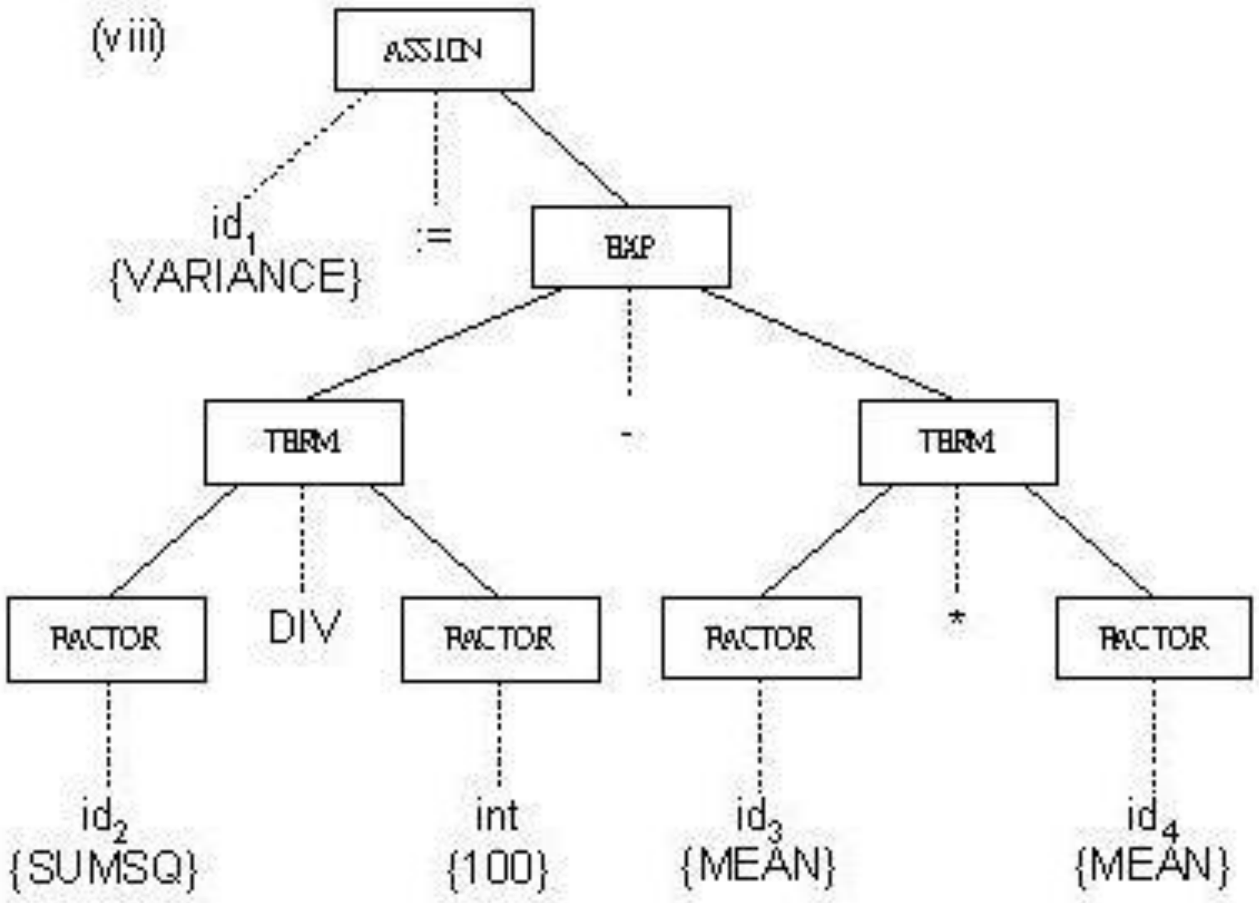

(v)
ASSIGN
id$_1$
{VARIANCE}
:=
EXP
TERM
FACTOR
DIV
FACTOR
id$_2$
{SUMSQ}
int
{100}

(vi)


ASSIGN
id_1 {VARIANCE}
:=
EXP
TERM
-
TERM
FACTOR
DIV
FACTOR
id_2 {SUMSQ}
int {100}

(vii)


ASSIGN
id_1 {VARIANCE}
:=
EXP
TERM
-
TERM
FACTOR
DIV
FACTOR
FACTOR
id_2 {SUMSQ}
int {100}
id_3 {MEAN}


(viii)
ASSIGN
id1
{VARIANCE}
:=
EXP
TERM
-
TERM
FACTOR
DIV
FACTOR
FACTOR
*
FACTOR
id2
{SUMSQ}
int
{100}
id3
{MEAN}
id4
{MEAN}