

► நோக்கம்

- ❖ ஒரு பிரச்சனையை பகுப்பாய்வு செய்தலும்
ஒரு பிரச்சனைக்கான நெறிமுறையை
உருவாக்கலும்

► பிரச்சனையை பகுப்பாய்வு செய்தல்

- ஒரு பிரச்சனையை பகுப்பாய்வு செய்யும் போது மூன்று விடயங்கள் நடைபெறும். அவையாவன உள்ளீடு, வெளியீடு ,முறைவழியாக்கல் என்பன ஆகும்.

- அப் பிரச்சனையை தீர்த்து பெறப்படும் பெறுபேறு வெளியீடு எனப்படும்.
அவ் வெளியீடு பெறுவதற்கு வழங்கப்படும் தரவுகள் உள்ளீடு எனப்படும்
உள்ளீட்டை வெளியீடாகப் பெறப்படும் செயன்முறை முறைவழியாக்கல் எனப்படும்.



❖ உதாரணம்

✓ வட்டம் ஒன்றின் ஆரை தரப்படும் போது சுற்றளவு பரப்பளவு என்பவற்றை காணல்

➤ தரவு

- வட்டமொன்றின் ஆரை உள்ளீடு
- சுற்றளவு பரப்பளவு வெளியீடு
- ஆரையை பயன்படுத்தி சுற்றளவு பரப்பளவு காணும் செயன்முறை முறைவழியாக்கம் எனப்படும்.

❖ உதாரணம்

- அஞ்சலில் இடுவதற்கு உகந்த கடிதத்தை தயாரித்தல்
- தரவு
 - உள்ளீடு- கடிதத்தை எழுதுவதற்கு உகந்த ஓர் தாள், பேனா, கடித உறை, முத்திரை , பிசின்
 - வெளியீடு- அஞ்சலில் இடுவதற்கு உகந்த கடிதம்

▪ முறைவழியாக்கம்-

- ✓ கடிதத்தை எழுதுதல்
- ✓ கடிதத்தை மடித்து கடித உறையில் இடுதல்
- ✓ கடிதவுறையை ஒட்டுதல்
- ✓ கடிதவுறையில் முகவரியை எழுதுதல்
- ✓ முத்திரையை ஒட்டுதல்

○ குறிப்பு-இங்கு படிமுறை 4, படிமுறை 5 என்பன ஒழுங்குமுறை மாற்றப்படினும் பெறுபேறு மாற்றமடையாது.

► நெறிமுறை

- நோக்கத்திற்கேற்ப பிரச்சனை ஒன்றிற்கான துல்லியமான தீர்வினை பெறும் வகையில் படிமுறைகளை சீரான ஒழுங்குமறையில் கூறுவதே நெறிமுறையாகும்.
- நெறிமுறையில் மூன்று அடிப்படை அம்சங்கள் காணப்படலாம்
 - வரிசைமுறை
 - தெரிவுமுறை
 - மீளச்செயல்

► வரிசைமுறை

- ஒரு பிரச்சனையை தீர்ப்பதற்கான படிமுறை ஆரம்பத்தில் இருந்து இறுதிவரை சீரான ஒழுங்கு முறையில் காணப்படல் வரிசை முறையாகும்

► தெரிவு முறை

- பிரச்சனையை தீர்ப்பதற்கான படிமுறை நிபந்தனை ஒன்றை அடிப்படையாக கொண்டு தெரிவுக்கு உள்ளாகும் போது அம் முறை தெரிவு முறை எனப்படும்.

► மீளச்செயல்முறை

- பிரச்சனையொன்றை தீர்ப்பதற்கான படிமுறைகள் ஒன்றோ/பலவோ குறித்தவொரு நிபந்தனையை நிறைவேற்றுவதற்காக மீண்டும் மீண்டும் இடம் பெறுதல் மீளச்செயல்முறை எனப்படும்.

► நெறிமுறையின் வகைகள்




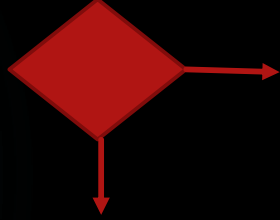
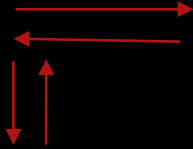

நெறிமுறையானது இரு வகைகளில் வெளிப்படுத்தமுடியும்

1. பாய்ச்சல் கோட்டுப்படம்
2. போலிக்குறிமுறை

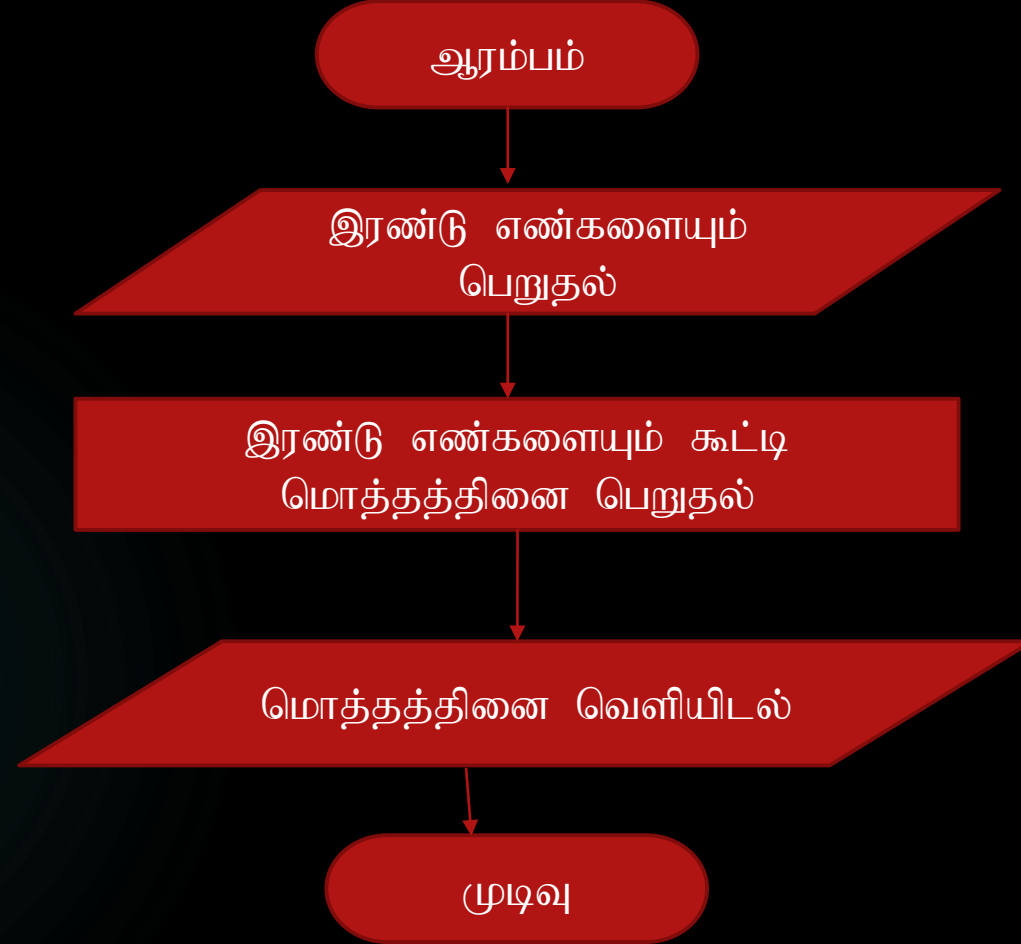
□ பாய்ச்சல் கோட்டுப்படம்

பிரச்சனைகளை தீர்ப்பதற்காக ஒழுங்குமுறையான படிமுறைகளை வரைபட குறியீட்டு ரீதியாக முன்வைப்பதே பாய்ச்சல் கோட்டு வரைபடமாகும்

► பாய்ச்சல் கோட்டுவரைபடத்தில் பயன்படுத்தப்படும் குறியீடுகளாவன

குறியீடு	பயன்பாடு
	தொடக்கம் அல்லது முடிவு
	உள்ளீடு அல்லது வெளியீடு
	முறைவழியாக்கம்
	தீர்மானம்
	பாய்ச்சல் திசை
	தொடர்பு

- ▶ இரண்டு எண்களைப் பெற்று அவற்றுக்கான கூட்டுத்தொகையை காண்பதற்கான நெறிமுறையை பாச்சற்கோட்டு படத்தில் தருக



□ போலிக்குறிமுறை

- பிரச்சனையை தீர்ப்பதற்கான படிமுறைகளை தமது சொந்த மொழியில் முன்வைத்தல் போலிக்குறிமுறை எனப்படும்.
 - இரண்டு எண்களைப் பெற்று அவற்றுக்கான கூட்டுத்தொகையை காண்பதற்கான போலிக்குறிமுறை
 - ✓ படிமுறை i- ஆரம்பம்
 - ✓ படிமுறை ii- இரண்டு எண்களையும் உள்வாங்கல்
 - ✓ படிமுறை iii- இரண்டு எண்களையும் கூட்டி மொத்தத்தை காணல்
 - ✓ படிமுறை iv- மொத்தத்தினை வெளியிடல்
 - ✓ படிமுறை v- முடிவு

பஸ்கால் செய்நிரல் மொழி

▶ அடையாளங்காணி (identifier) :-

அடையாளங்காணி என்பது ஒரு மாறியை அல்லது மாறிலியை அல்லது செய்நிரலை அறிமுகம் செய்யப்பயன்படுத்தப்படும் பெயர் ஆகும் இந்த அடையாளங்காணியைப் பயன்படுத்துகையில் பின்வரும் பொதுவான விதிகளை பயன்படுத்தவேண்டும்

- ✓ பஸ்கால் மொழியிலுள்ள ஒதுக்குசெய்சொற்களை பயன்படுத்தமுடியாது உதாரணமாக Var ,print, begin , end, const, Input , Output, Of , Read , Write, Readln , writeln,

► குறிப்பு-

- அடையாளங்காணியின் முதல் எழுத்து ஓர் ஆங்கில நெடுங்கணக்கில் உள்ள எழுத்தாக இருத்தல் வேண்டும்
- தொடர்ந்து ஆங்கில எழுத்துக்களோ இலக்கங்களோ வரலாம்.
- இங்கு under score இனைத் தவிர வேறு எந்தவொரு வரியுருக்களும் இடம்பெறக்கூடாது (! # \$ & *?/)
- மேலும் சொற்களுக்கிடையே இடைவெளி இடம்பெறக்கூடாது

❖ சரியான அடையாளங்காணிக்கு உதாரணம்

- Sum
- Total_No
- F_Name
- No1
- No2

❖ பிழையான அடையாளங்காணிக்கு உதாரணம்

- 6Sum
- Student Name
- Last-Name
- \$Average

❖ இங்கு எழுத்து பேதம் பொருட்படுத்தப்படுவதில்லை

உதாரணம்- Sum,SUM ஒரே அடையாளங்காணியாகும்.

► Data type

- Integer-நேர் அல்லது மறை நிறையெண்கள்
- Real- Integer-நேர் அல்லது மறைமெய்யெண்கள்
- Boolean-True அல்லது False
- Char-சாவிப்பலகையில் உள்ள யாதாயினும் ஒரு வரியுரு உதாரணம் 's',\$,9
- String-சாவிப்பலகையில் உள்ள யாதாயினுமொரு வரியுருத்தொடரி
உதாரணம்-F_Name,school

► எந்தவொரு செய்நிரல் மொழியிலும் அத்தரவினை தேக்கிவைப்பதற்கு தேவையான இடத்தின் அளவு தீர்மானிக்கப்படுகின்றது. ஆகவே தரவு வகையினை பற்றி செய்நிரலாளர் அறிந்திருத்தல் அவசியமாகின்றது

► குறிப்பு

Char,String எனும் தரவு வகைகளின் பெறுமதிகள் ஒற்றை மேற்கோட்குறிக்குள் குறிப்பிடப்படவேண்டும்.

► மாறிகளும் மாறிலிகளும்

□ மாறிகள்

மாறிகள் ஒருவகை அடையாளங்காணியாகும். செய்நிரல் நடைமுறைப்படுத்தப்படும் போது மாறிக்கு ஒதுக்கப்படுகின்ற பெறுமானங்கள் மாறிக்கொண்டிருக்கும். இங்கு மாறியை வகைகுறிக்க Var எனும் ஒதுக்குசொல் பயன்படுத்தப்படுகின்றது

உதாரணம்

```
Var count : integer
```

```
    a,b : real
```

```
    Pass : boolean
```

□ மாறிலி

செய்நிரல் நடைமுறைப்படுத்தும் போது அடையாளங்காணிக்கு ஒதுக்கப்படுகின்ற பெறுமானம் மாற்றமடையாது இருப்பின் அத்தகைய அடையாணங்காணி மாறிலி எனப்படும். இது Const எனும் ஒதுக்குச் சொல்லினால் வகை குறிக்கப்படுகின்றது.

$$\text{ஊதாரணம் ; - } \text{Pi} := \frac{22}{7}$$

வட்டத்தின் ஆரை தரப்படும் போது வட்டத்தின் சுற்றளவு பரப்பளவு காணப்படும் செய்நிரலில் $\text{Pi} := \frac{22}{7}$ என்பது எப்போதும் மாறாமல் இருக்கும்.

► வினைக்குறிகள்

1. எண்கணித வினைக்குறிகள்

வினைக்குறிகள்	பயன்பாடு
+	கூட்டல்
-	கழித்தல்
*	பெருக்கல்
/	வகுத்தல்
DIV	வகுத்தலின் பின் ஈவு
MOD	வகுத்தலின் பின் மீதி

உதாரணம்-

$$10 \text{ DIV } 3=3$$

$$10 \text{ MOD } 3=1$$

2. ஒப்பீட்டு வினைக்குறிகள்

வினைக்குறிகள் பயன்பாடு

> பெரியது

>= பெரியது அல்லது சமன்

< சிறியது

<= சிறியது அல்லது சமன்

= சமன்

<> சமனற்றது

3. தர்க்க வினைக்குறி

a. AND வினைக்குறி

(முதலாம் கோவை) AND (இரண்டாம் கோவை) இல் முதலாம் கோவையும் இரண்டாம் கோவையும் உண்மையாக இருக்கும் சந்தர்ப்பத்தில் பெறுபேறு உண்மையாகவும் மற்றைய சந்தர்ப்பங்களில் பெறுபேறு பொய்யாகவும் இருக்கும்.

ஊதாரணம்

$(3 \geq 2)$ AND $(4 < 4)$ எனும் உதாரணத்தில்

$(3 \geq 2)$ என்பது உண்மை

$(4 < 4)$ என்பது பொய்யாகும்

ஆகவே முடிவு பொய்யாகும்

ஊதாரணம்

$(3 \geq 2)$ AND $(4=4)$ எனும் உதாரணத்தில்

$(3 \geq 2)$ என்பது உண்மை

$(4=4)$ என்பதும் உண்மை

ஆகவே முடிவு உண்மையாகும்

ஊதாரணம்

$(3=2)$ AND $(5>4)$ எனும் உதாரணத்தில்

$(3=2)$ என்பது பொய்

$(5>4)$ என்பது உண்மை

ஆகவே முடிவு பொய்யாகும்

ஊதாரணம்

$(3=2)$ AND $(4 \neq 4)$ எனும் உதாரணத்தில்

$(3=2)$ என்பது பொய்

$(4 \neq 4)$ என்பது பொய்

ஆகவே முடிவு பொய்யாகும்

► b) OR வினைக்குறி

(முதலாம் கோவை) AND (இரண்டாம் கோவை) இல் முதலாம் கோவையும் இரண்டாம் கோவையும் பொய்யாக இருக்கும் சந்தர்ப்பத்தில் பெறுபேறு பொய்யாகவும் மற்றைய சந்தர்ப்பங்களில் பெறுபேறு உண்மையாகவும் இருக்கும் உதாரணம்

$(3 > 2)$ OR $(4 = 4)$ எனும் உதாரணத்தில்

$(3 > 2)$ என்பது உண்மை

$(4 = 4)$ என்பது உண்மை

ஆகவே முடிவு உண்மையாகும்

(3>=2) OR (4<>4) எனும் உதாரணத்தில்

(3>=2) என்பது உண்மை

(4<>4) என்பது பொய்யாகும்

ஆகவே முடிவு உண்மையாகும்

(3=2) OR (4=4) எனும் உதாரணத்தில்

(3=2) என்பது பொய்

(4= 4) என்பது உண்மை

ஆகவே முடிவு உண்மையாகும்

(3=2) OR (4<>4) எனும் உதாரணத்தில்

(3=2) என்பது பொய்

(4<>4) என்பது பொய்

ஆகவே முடிவு பொய்யாகும்

C) NOT வினைக்குறி

NOT வினைக்குறியானது உண்மையான கோவையொன்றினை பொய்யாகவும் பொய்யான கோவையொன்றினை உண்மையாகவும் காட்டும்.

ஊதாரணம் $(3 \geq 2)$ என்பது உண்மையாகும்

$\text{NOT}(3 \geq 2)$ என்பது பொய்யாகும்

ஊதாரணம் $(3 = 2)$ என்பது பொய்

$\text{NOT}(3 = 2)$ என்பது உண்மையாகும்

பஸ்கால் செய்நிரல்மொழி

▶ இரண்டு எண்கள் தரப்படும் பொழுது அவ்விரண்டு எண்களையும் கூட்டி மொத்தத்தினை வெளியிடுவதற்கான செய்நிரலை கருதுவோம்

▶ Program addNum(Input,output);

```
var num1,num2,total : integer;
```

```
Begin
```

```
write('Welcome Pascal Programming');
```

```
{calculate the total of two numbers}
```

```
Writeln('Enter first Number');
```

```
Read(num1);
```

```
Writeln('Enter second Number');
```

```
Read(num2);
```

```
Total:=num1+num2;
```

```
writeln('Total is =' total);
```

```
End.
```

ஒரு வட்டத்தின் பரப்பளவை கணிப்பதற்கான நெறிமுறையினை
போலிக்குறிமுறையை பயன்படுத்தி காண்க

படிமுறை1 ஆரம்பம்

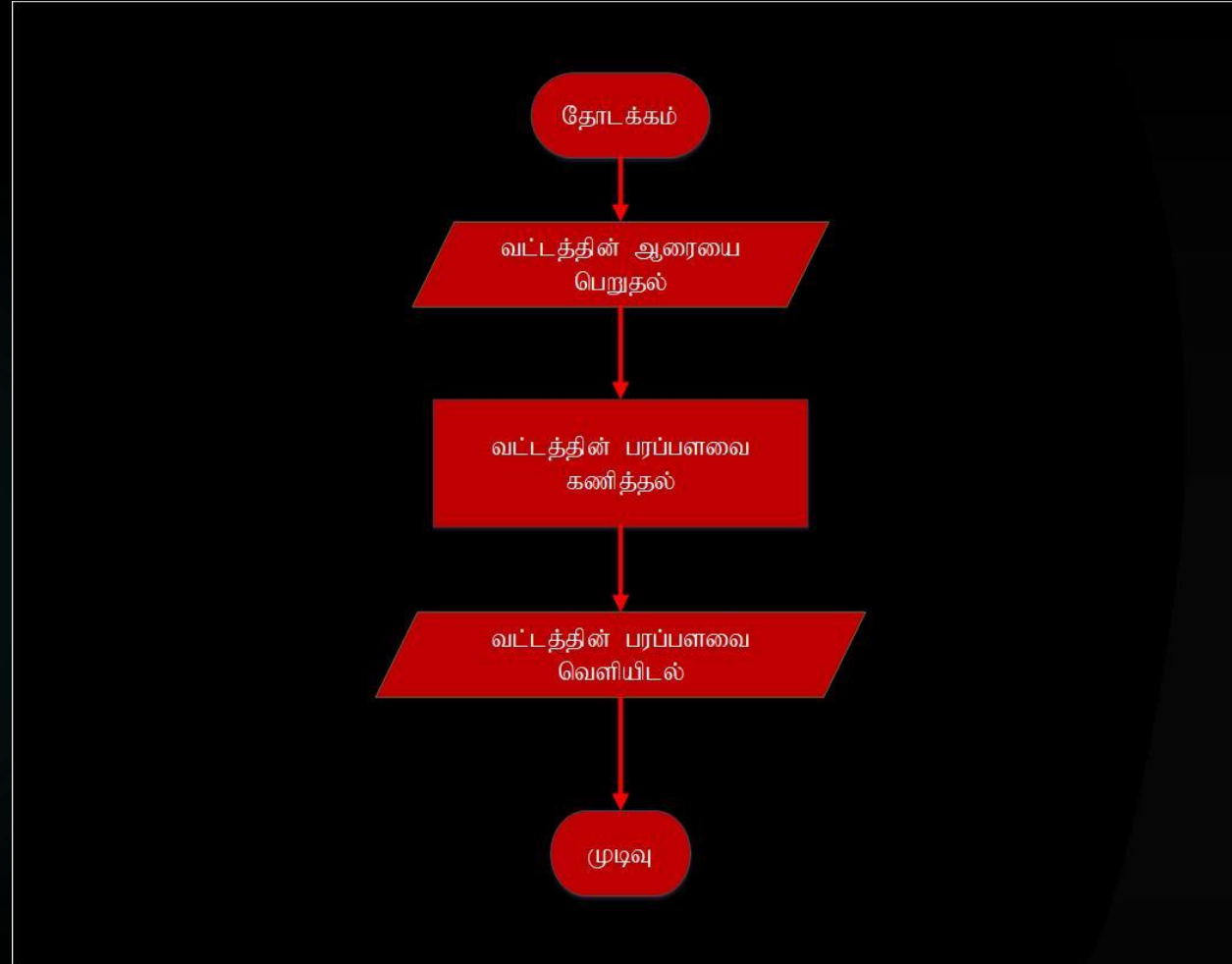
படிமுறை2 வட்டத்தின் ஆரை r யை பெறுதல்

படிமுறை3 வட்டத்தின் பரப்பளவு $A=\pi*r*r$ எனும் சமன்பாட்டை பயன்னடுத்தி
காணல் (இங்கு $\pi==22/7$)

படிமுறை4 வட்டத்தின் பரப்பளவை வெளியிடல்

படிமுறை 5 முடிவு

ஒரு வட்டத்தின் பரப்பளவை கணிப்பதற்கான நெறிமுறையினை
பாச்சற்கோட்டுப்படத்தை பயன்படுத்தி காண்க



ஒரு வட்டதின் பரப்பளவை கணிப்பதற்கான செய்நிரலை
பஸ்கால் மொழியில் தருக

▶ Program area(input,output);

```
Var   r : Integer;
```

```
      A: real;
```

```
Const pi=22/7;
```

```
begin
```

```
    writeln('Enter Circle radius');
```

```
    read(r);
```

```
    A:=pi*r*r;
```

```
    writeln('Circle Area =',A);
```

```
End.
```

பெறப்படும் எண் நேரெண் எனின் மாத்திரம் அந்த எண்ணை வெளியிடுவதற்கான நெறிமுறையினை போலிக்குறிமுறையில் தருக

▶ Begin

input N

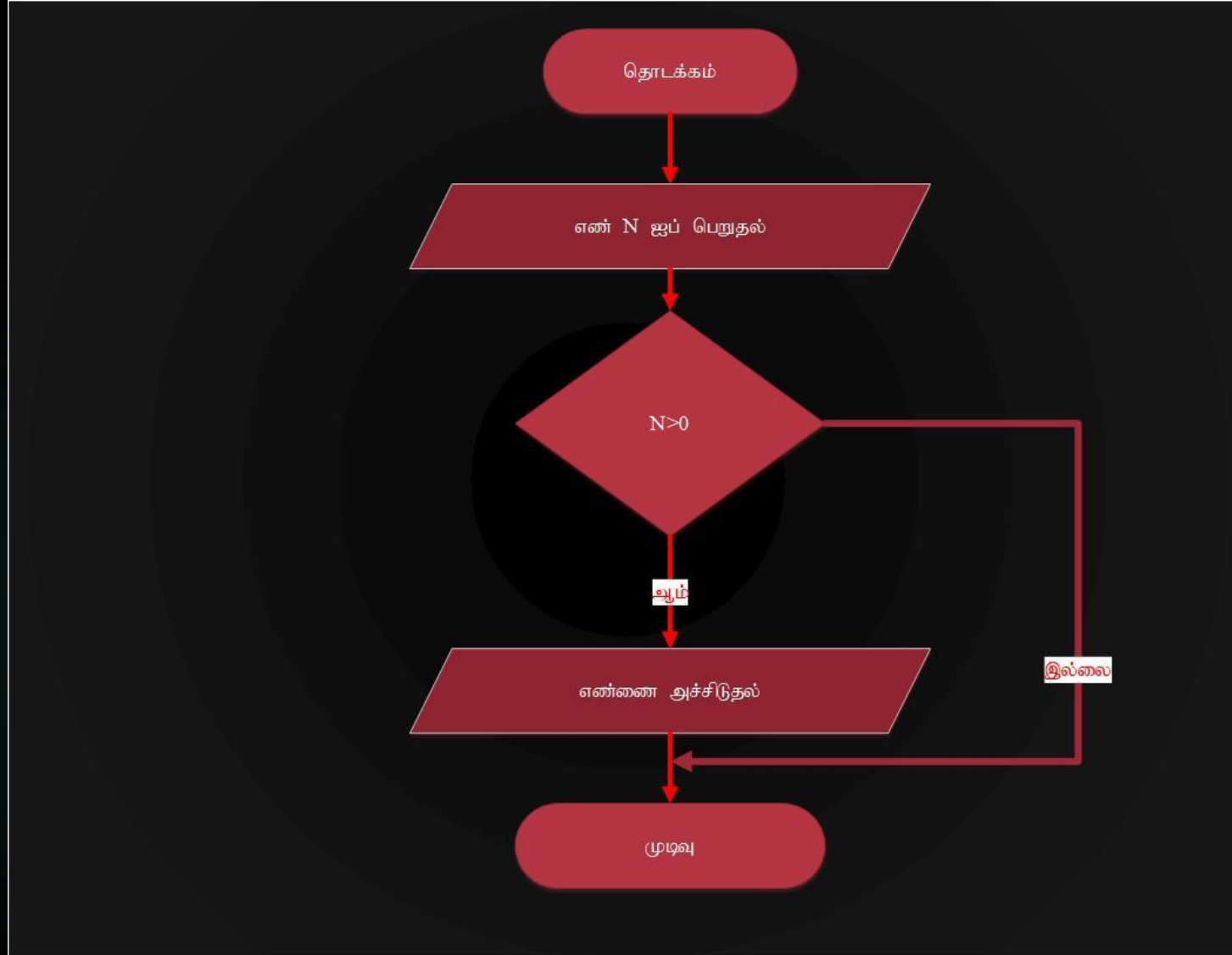
IF N>0 Then

print N

Endif

End.

பெறப்படும் எண் நேரெண் எனின் மாத்திரம் அந்த எண்ணை வெளியிடுவதற்கான நெறிமுறையினை பாச்சற்கோட்டுப்படத்தினை பயன்படுத்தி தருக



பெறப்படும் எண் நேரெண் எனின் மாத்திரம் அந்த எண்ணை வெளியிடுவதற்கான செய்நிரலை பஸ்கால் மொழியில் தருக

► Program positiveNo(Input,output);

```
Var    N: integer;
```

```
Begin
```

```
    writeln('Enter Number');
```

```
    Read(N);
```

```
    If N>0 then
```

```
        writeln(N);
```

```
END.
```


சமமற்ற இரு எண்கள் தரப்படும்பொழுது அவற்றுள் மிகப்பெரிய எண்ணை காண்பதற்கான நெறிமுறையினை போலிக்குறிமுறையில் தருக

▶ Begin

Input n1,n2

If $n1 > n2$ then

Large=n1


Else

Large=n2

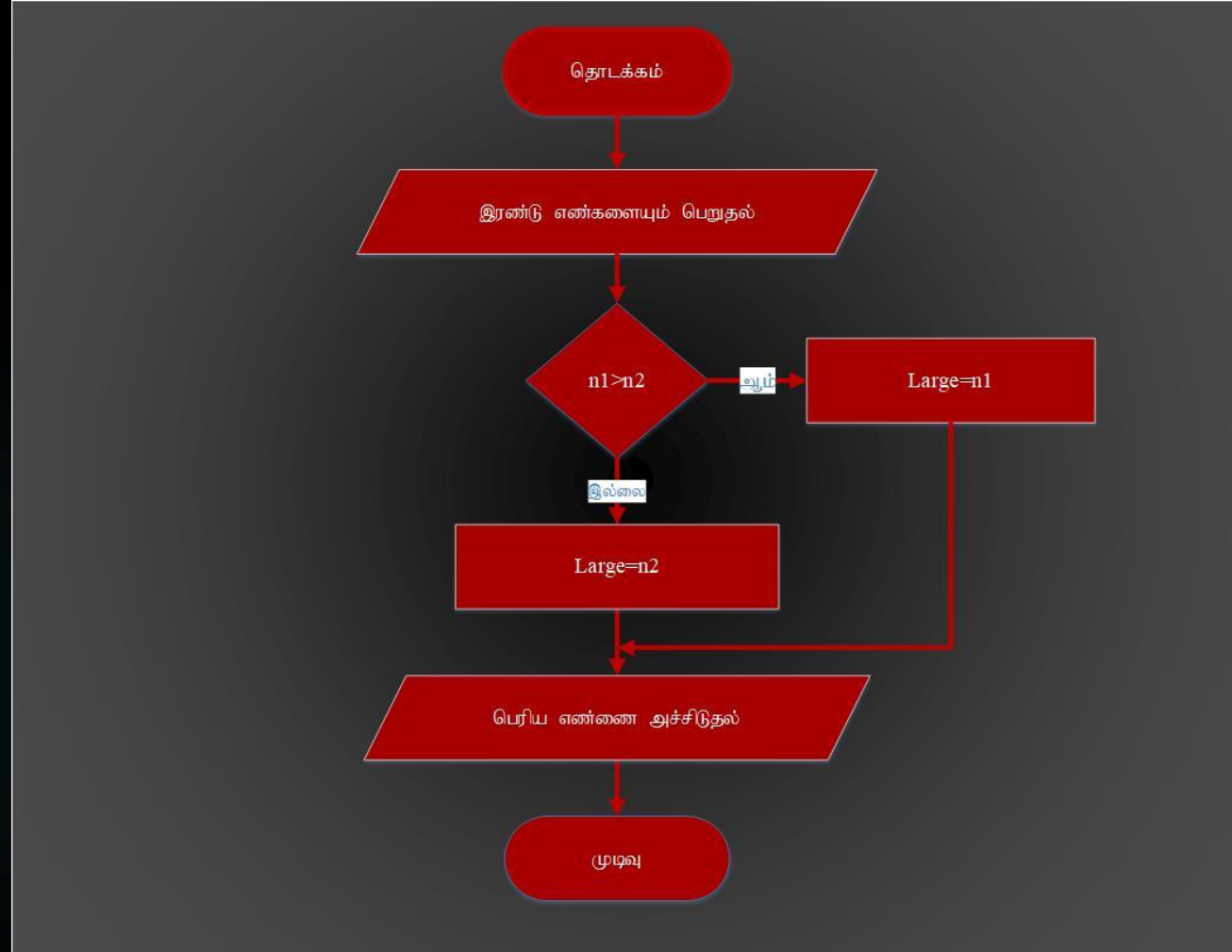
EndIf

Print Large

End

- 
- ▶ [Lazarus-2.0.8-fpc-3.0.4 -win64](#)
 - ▶ Lazarus is a free cross-platform visual integrated development environment (IDE) for rapid application development using the Free Pascal compiler.
 - ▶ <https://sourceforge.net/projects/lazarus/>

சமமற்ற இரு எண்கள் தரப்படும்பொழுது அவற்றுள் மிகப்பெரிய எண்ணை காண்பதற்கான நெறிமுறையினை பாச்சற்கோட்டுபடத்தினை பயன்படுத்தி தருக



சமமற்ற இரு எண்கள் தரப்படும்பொழுது அவற்றுள் மிகப்பெரிய எண்ணை காண்பதற்கான செயல்நிரலினை பஸ்கால் மொழியில் தருக

▶ Program LargeNo(input,output);

```
Var    n1,n2,Large : Integer;
```

```
Begin
```

```
    Writeln('Enter Two Numbers');
```

```
    Read(n1,n2);
```

```
    If n1>n2 then
```

```
        Large=n1
```

```
    Else
```

```
        Large=n2;
```

```
    writeln('Large Number is=',Large);
```

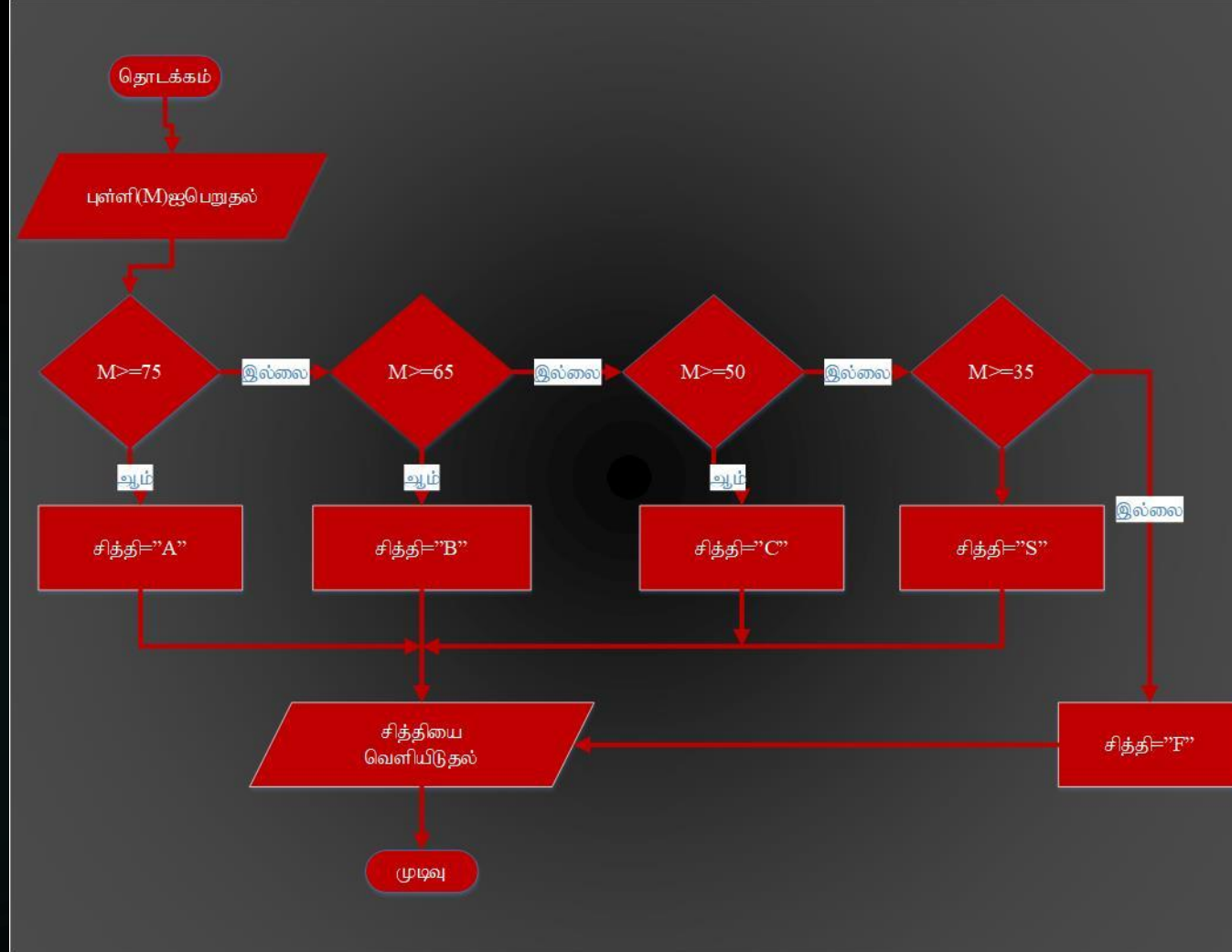
```
End.
```

NESTED IF

மாணவனொருவன் பாடமொன்றிற்கு பெற்றபுள்ளிகளை உள்ளிடும்போது அதற்கான சித்தியைக்காண்பதற்கான நெறிமுறையினை போலிக்குறிமுறையில் தருக

```
▶ Begin
    Input Marks (M)
    IF M>=75 Then
        Grade='A'
    Else
        IF M>= 65 Then
            Grade='B'
        Else
            IF M>=50 Then
                Grade='C'
            Else
                IF M>=35 Then
                    Grade='S'
                Else
                    Grade ='F'
                EndIf
            EndIf
        EndIf
    EndIf
    Display Grade
End
```

மாணவனொருவன் பாடமொன்றிற்கு பெற்றபுள்ளிகளை உள்ளிடும்போது அதற்கான சித்தியைக்காண்பதற்கான நெறிமுறையினை பாச்சற்கோட்டுப்படத்தில் தருக



மாணவனொருவன் பாடமொன்றிற்கு பெற்றபுள்ளிகளை உள்ளிடும்போது அதற்கான சித்தியைக்காண்பதற்கான செய்நிரலினை பஸ்கால் மொழியில் எடுத்துரைக்க

► Program GradeForMarks(Input,output);

```
Var M : Inteher;
```

```
Grad : Char;
```

```
Begin
```

```
Writeln('Enter Marks');
```

```
Read(M);
```

```
If M >= 75 then
```

```
    Grade := 'A'
```

```
Else
```

```
    If M >= 65 then
```

```
        Grade := 'B'
```

```
    Else
```

```
        If M >= 50 then
```

```
            Grade := 'C'
```

```
    Else
```

```
        If M >= 35 then
```

```
            Grade := 'S'
```

```
        Else
```

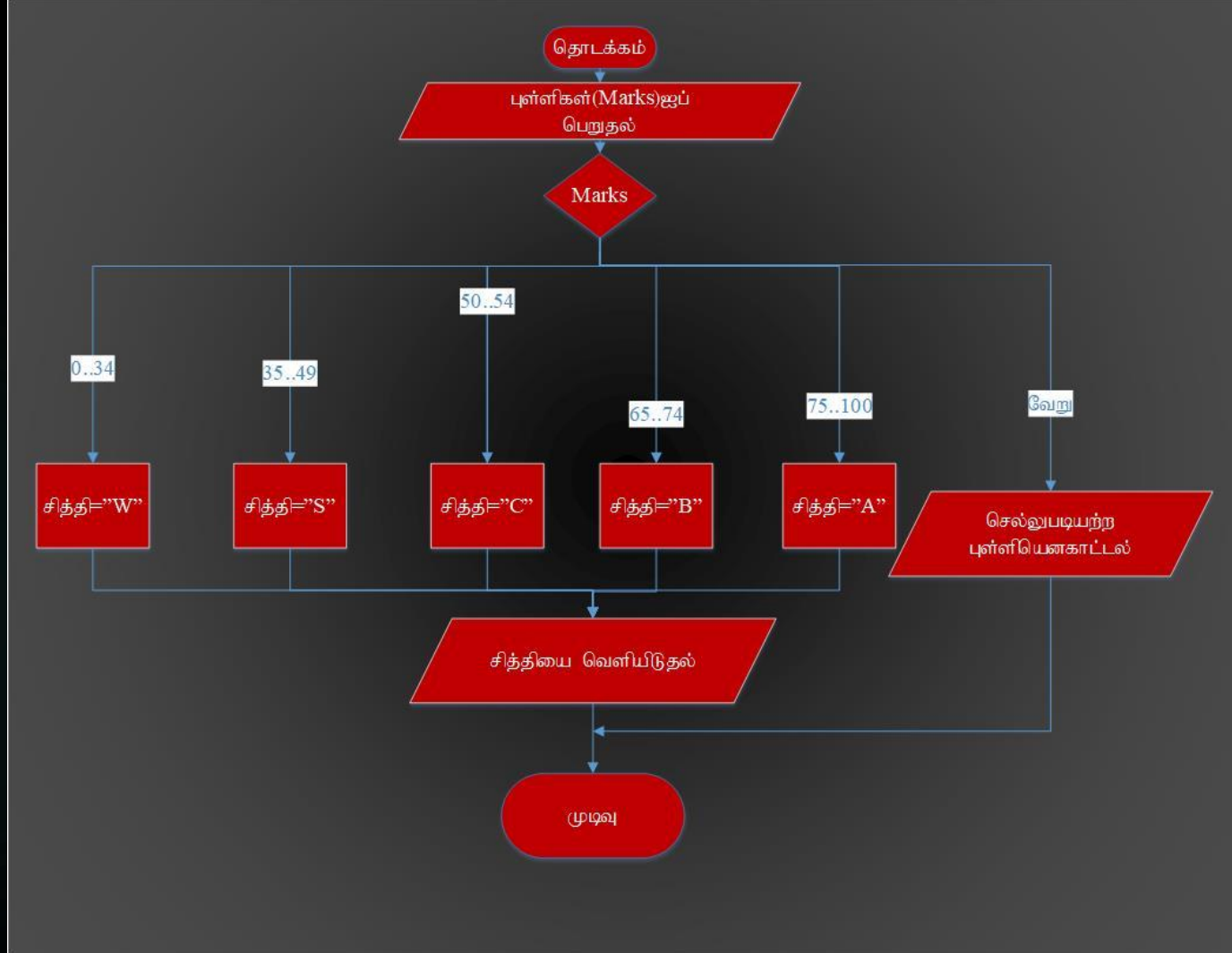
```
            Grade := 'F';
```

```
Writeln('Grade= ', Grade);
```

```
End.
```

Case Statemnet

மாணவனொருவன் பாடமொன்றிற்கு பெற்றபுள்ளிகளை உள்ளிடும்போது அதற்கான சித்தியைக்காண்பதற்கான நெறிமுறையினை பாச்சற்கோட்டுப்படத்தில் தருக



மாணவனொருவன் பாடமொன்றிற்கு பெற்றபுள்ளிகளை உள்ளிடும்போது அதற்கான சித்தியைக்காண்பதற்கான செய்நிரலினை பஸ்கால் மொழியில் எடுத்துரைக்க

► Program Findgrade(Input,Output);

Var Marks : Integer;

Grade : Char;

Begin

Writeln('Enter Marks');

Read(Marks);

Case Marks of

0..34 : Grade:='F';

35..49 : Grade:='S';

50..64 : Grade:='C';

65..74 : Grade:='B';

75..100 : Grade:='A';

Else

writeln('Invalid marks');

End;

If (Marks>=0) AND (Marks<=100) then

writeln('Grade is=', Grade);

End.

Mr.P.Thavanathan

ISA - ICT

Zonal ICT Education Centre

Grade 11 - ICT

பிரச்சனைகளைத் தீர்ப்பதற்குச்

செய்நிரல்களை எழுதுதல்

தரம்11 அலகு 1

மீள்செயல்களைப்பயன்படுத்தி செய்நிரல்களை எழுதுதல்

- ▶ 1) FOR – DO (கட்டமைப்பு முறை 1)

FOR Variable:= value_1 TO Value_2 DO

ஊதாரணம் : 1 தொடக்கம் 5 வரையுள்ள

பெறுமானங்களை வெளியிடல்

```
Program PrintNos(Input,output);
```

```
Var Count : Integer;
```

```
Begin
```

```
For Count := 1 to 5 do
```

```
writeln(Count);
```

```
Readln;
```

```
End.
```

- ▶ வெளியீடு

- ▶ 1

- ▶ 2

- ▶ 3

- ▶ 4

- ▶ 5

► FOR – DO (கட்டமைப்பு முறை 2)

```
FOR Variable:= value_1 DOWNTO Value_2 DO
```

ஊதாரணம் : 1 தொடக்கம் 10 வரையுள்ள
பெறுமாணங்களை வெளியிடல்

```
Program PrintNos(Input,output);
```

```
Var Count : Integer;
```

```
Begin
```

```
For Count := 10 downto 5 do
```

```
writeln(Count);
```

```
Readln;
```

```
End.
```

► வெளியீடு

► 10

► 9

► 8

► 7

► 6

► 5

While do கட்டமைப்பு

கட்டமைப்பு While Condition do

ஊதாரணம் : 1 தொடக்கம் 5 வரையுள்ள
பெறுமானங்களை வெளியிடல்

```
Program PrintNos(Input,output);
```

```
Var Count : Integer;
```

```
Begin
```

```
    Count :=1
```

```
    while Count <=5 do
```

```
        writeln(Count);
```

```
    Readln;
```

```
End.
```

▶ வெளியீடு

▶ 1

▶ 2

▶ 3

▶ 4

▶ 5

கட்டமைப்பு While Condition do

ஊதாரணம் : 1 தொடக்கம் 4 வரையுள்ள பெறுமாணங்களை வெளியிடல்

```
Program PrintNos(Input,output);
```

```
Var Count : Integer;
```

```
Begin
```

```
    Count :=1
```

```
    while Count <5 do
```

```
        writeln(Count);
```

```
    Readln;
```

```
End.
```

▶ வெளியீடு

▶ 1

▶ 2

▶ 3

▶ 4

Repeat Until கட்டமைப்பு

▶ கட்டமைமைப்பு

Repeat

Statement/Statements

Until Condition

உதாரணம் : 0 தொடக்கம் 5 வரையுள்ள பெறுமாணங்களை வெளியிடல்

Program repeataction();

Var Count: Integer;

Begin

Count=0;

Repeat

writeln(Count);

Count:=Count+1;

Until Count>5

End.

▶ வெளியீடு

▶ 0

▶ 1

▶ 2

▶ 3

▶ 4

▶ 5

உதாரணம் : 0 தொடக்கம் 4 வரையுள்ள
பெறுமானங்களை வெளியிடல்

```
Program repeataction( );
```

```
Var    Count: Integer;
```

```
Begin
```

```
    Count=0;
```

```
    Repeat
```

```
        writeln(Count);
```

```
        Count:=Count+1;
```

```
    Until Count>=5
```

```
    Readln;
```

```
End.
```

▶ வெளியீடு

▶ 0

▶ 1

▶ 2

▶ 3

▶ 4

```
Program repeataction( );
```

```
Var    Count: Integer;
```

```
Begin
```

```
    Count=0;
```

```
    Repeat
```

```
        writeln(Count);
```

```
        Count:=Count+1;
```

```
    Until Count<5
```

```
    Readln;
```

```
End.
```

▶ வெளியீடு

▶ 0

அணிகளை பயன்படுத்தல்

- ▶ செய்நிரலில் ஒரேவகையான பலதரவுகளை சேமித்துவைத்து செய்நிரல் செயற்படுத்தப்படும்போது பயன்படுத்த அணிகள் பயன்படுத்தபடுகின்றன.

உதாரணமாக ஆண்டு 12 இல் கணிததுறையில் கல்விகற்கும் மாணவனொருவன் இரண்டாம் தவணை பரிச்சையில் பெற்றுக்கொண்ட புள்ளிகளுக்கிணங்க முன்று பாடங்களிலும் மாணவன் பெற்ற புள்ளிகளின் கூட்டுதொகை , சராசரி என்பவற்றை காண்போம். இங்கு செயல்நிரல் செயற்படுத்தப்படும்போது மாணவன் முன்றுபாடத்திற்கும் பெற்ற புள்ளி சேமித்து வைத்து பயன்படுத்தவேண்டியிருக்கும். இதற்காக பின்வருமாறு அணி வரையறுக்கப்படவேண்டும்.

```
Var Marks: Array[0..2] of integer
```

M[0]

M[1]

M[2]

67	98	43
----	----	----

▶ Program grade12(Input,output);

```
Var M: Array[0..2] of Integer;
```

```
    i, total :Integer;
```

```
    avg:real;
```

```
begin
```

```
    total:=0;
```

```
    avg:=0;
```

```
    for i:=0 to 2 do
```

```
        begin
```

```
            Writeln('Enter Marks');
```

```
            Readln(M[i]);
```

```
            total:=total+M[i];
```

```
        End;
```

```
    avg:=total/3;
```

```
    writeln('Total Marks=' ,total);
```

```
    writeln('Average=' , avg);
```

```
Readln;
```

```
End.
```

உபசெய்நிரல்களை பயன்படுத்தல்

- ▶ சேய்நரல்கள் எழுதப்படுகின்றபொழுது அச்செய்நிரல்கள் மிகநீண்டதாக இருக்கும்போது அதனைவாசித்து விளங்கிகொள்ளல் மற்றும் செய்நிரலைபேணல் மிகவும் கடினமானதாகும். இதனால் அச்செயல்நிரலை இயன்றவரை பல உபசெயல்நிரல்களாக எழுதுதல் அதனை வாசிக்க மற்றும் பேண இலகுவானதாகவிருக்கும்.

உபசெய்நிரல்வகைகள்

- ▶ பிரதானசெய்நிரலுடன் தொடர்புபட்ட உபசெய்நிரல்கள் மூலம் பிரதானசெய்நிரலுக்கு வெளியீட்டை திரும்பவழங்கத்தக்க , வழங்கமுடியாத இருவகை உபசெய்நிரல்கள் உள்ளன. வெளியீட்டை திரும்பவழங்கத்தக்க செய்நிரல்கள் சார்புகள் (Function) எனவும் வெளியீட்டை திரும்பவழங்கமுடியாத செய்நிரல்கள் நடைமுறை (Procedure) எனவும் அழைக்கப்படும்.

நடைமுறையை அறிமுகம்செய்யும்முறை

▶ Procedure Name_of_procedure(name_of_variable:data type)

▶ உதாரணம்

ஒருவட்டத்தின் பரப்பளவை காண்பதற்குரிய தீர்வின்போது நடைமுறை பின்வருமாறு வரையறுக்கப்பயன்படுகின்றது.

```
Procedure calculateArea(Var radius : real);
```


சார்பை அறிமுகம்செய்யும்முறை

▶ Function Name_of_function(name_of_variable:data type) :data type;

▶ உதாரணம்

ஒருவட்டத்தின் பரப்பளவை காண்பதற்குரிய தீர்வின்போது சார்பு பின்வருமாறு வரையறுக்கப்பயன்படுகின்றது.

```
Function calculateArea(Var radius : real):real;
```

▶ ஓர்வட்டத்தின் ஆரைதரப்படும்பொழுது பரப்பளவையும் சுற்றளவையும் கணிப்பதற்கான செய்நிரலை

1) நடைமுறையை பயன்படுத்தி

2) சார்பை பயன்படுத்தி

தருக.

```
Program Pro_Circle(Input,output);  
Const pi=22/7;  
Var r:real;  
Procedure getdata(Var r:real);  
begin  
writeln('Enter Radius of the circle');  
readln(r);  
end;
```

```
► Procedure computeArea(Var r:real);
  Var A:real;
  Begin
    A:=Pi*r*r;
    Writeln('The Area of the Circle=',A);
  End;
Procedure Computecircumference(Var r:real);
  Var C:real;
  Begin
    C:=2*pi*r
    Writeln('The circumference of the circle=',C);
  End;
Begin
  getdata(r);
  computeArea(r);
  Computecircumference(r);
End.
```

```
► Program Fun_Circle(Input,Output);  
  
const pi=22/7;  
  
Var r:real;  
  
function computeArea(Var r:real):real;  
  Var A:real;  
  Begin  
    A:=pi*r*r;  
    computeArea:=A;  
  End;  
  
function computecircumference(Var r:real):real;  
  Var C:real;  
  Begin  
    C:=2*pi*r;  
    computecircumference:=C;  
  End;
```

```
Begin  
  Writeln('Enter Radius');  
  Readln(r);  
  Writeln('Circumference of the circle=', computecircumference(r));  
  Writeln('Area of the Circle=', computeArea(r));  
  Readln;  
End.
```