No :   Date: /   /

PYTHON

Scanned by CamScanner

29/11/2018

පරිගණක ඇසුරෙන් ගැටලු විසඳීම. (Python)

◂ ගැටලුවක් විසඳීමේ පියවර

* ගැටලුව හඳුනා ගැනීම.
* විකල්ප විසඳුම් යෝජනා කිරීම.
* විකල්ප විසඳුම් අතරින් වඩාත් ගැලපෙන විසඳුම තෝරා ගැනීම
* ඒ සඳහා ඇල්ගොරිතමයක් ගොඩනැගීම.
* සුදුසු පරිගණක භාෂාවකින් කේතනය කිරීම.
* පරීක්ෂා කිරීම.
* ක්‍රියාත්මක කිරීම.

* ඇල්ගොරිතමය

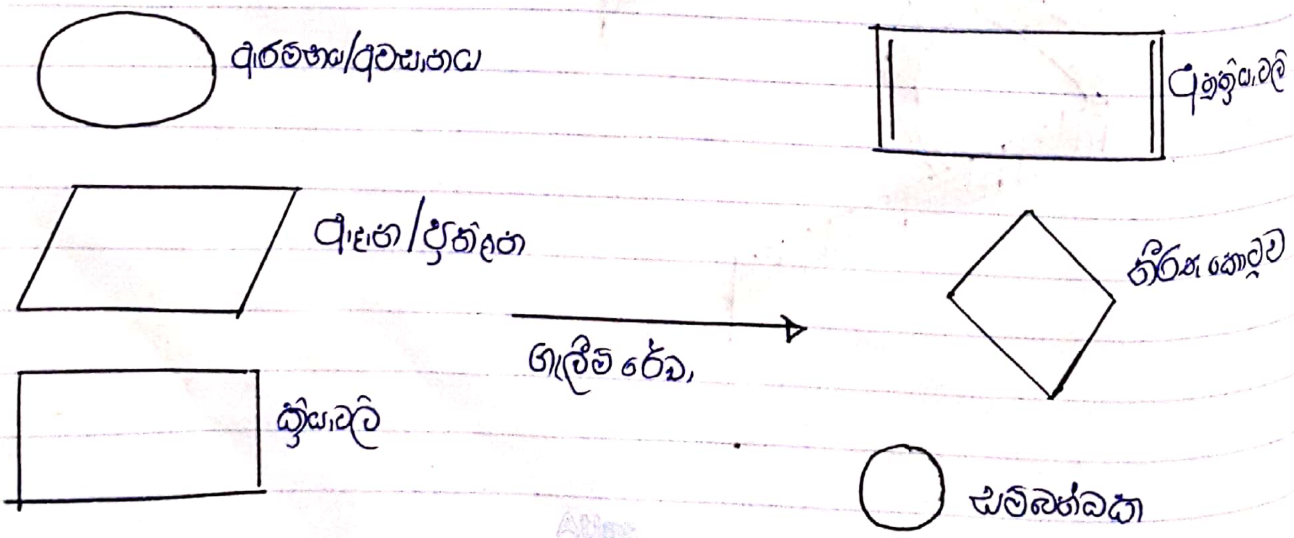
ඇල්ගොරිතමයක් යනු, කිසියම් ගැටලුවක් විසඳීමේ පියවර අනුපිළිවෙලින් දැක්වීමයි.
ඇල්ගොරිදමයක් ඉදිරිපත් කිරීම සඳහා ක්‍රමවේද 02ක් භාවිත කරයි.

01. ගැලීම් සටහන්
02. ව්‍යාජකේත

01. ගැලීම් සටහන්

ඇල්ගොරිදමයක් චිත්‍රක නිරූපණය කිරීම සඳහා ගැලීම් සටහන් භාවිත කරනු ලබයි.
ඒ සඳහා සංකේත භාවිත කරනු ලබයි.

◂ ගැලීම් සටහනක භාවිත වන සංකේත



Scanned by CamScanner

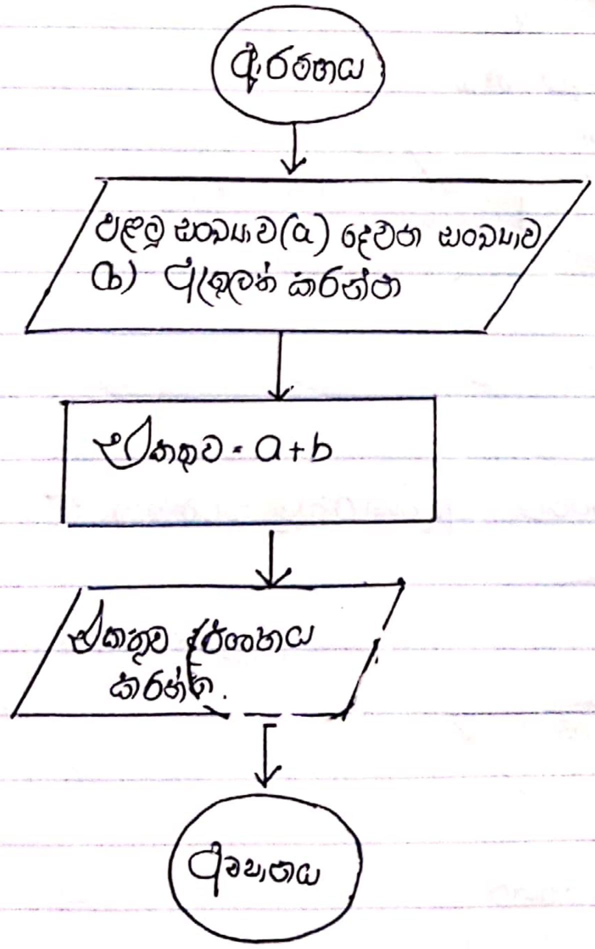
ඇල්ගොරිදමයක පාලන ව්‍යුහ ආකාර 03 කි.

01. අනුක්‍රම පාලන ව්‍යුහය
02. තේරීම් පාලන ව්‍යුහය
03. පුනරාවර්තන පාලන ව්‍යුහය

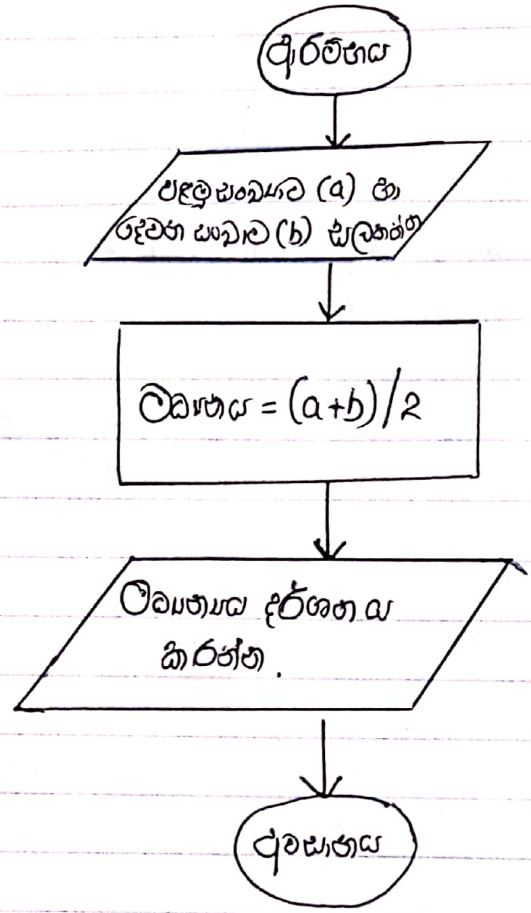
• ~~තේරීම්~~ අනුක්‍රම පාලන ව්‍යුහයක් සහිත ගැලීම් සටහන්

ඇල්ගොරිතමයේ පියවර එකකට පසු එකක් පිළිවෙලින් ගලායයි නම්, එහි පාලන ව්‍යුහය "අනුක්‍රම පාලන ව්‍යුහය" වේ.

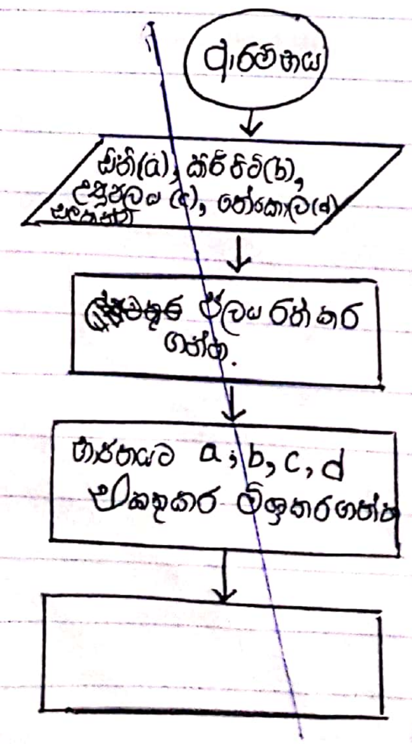
• සංඛ්‍යා 02ක් ලබාගෙන එම සංඛ්‍යා 02හි එකතුව ප්‍රදර්ශනය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.



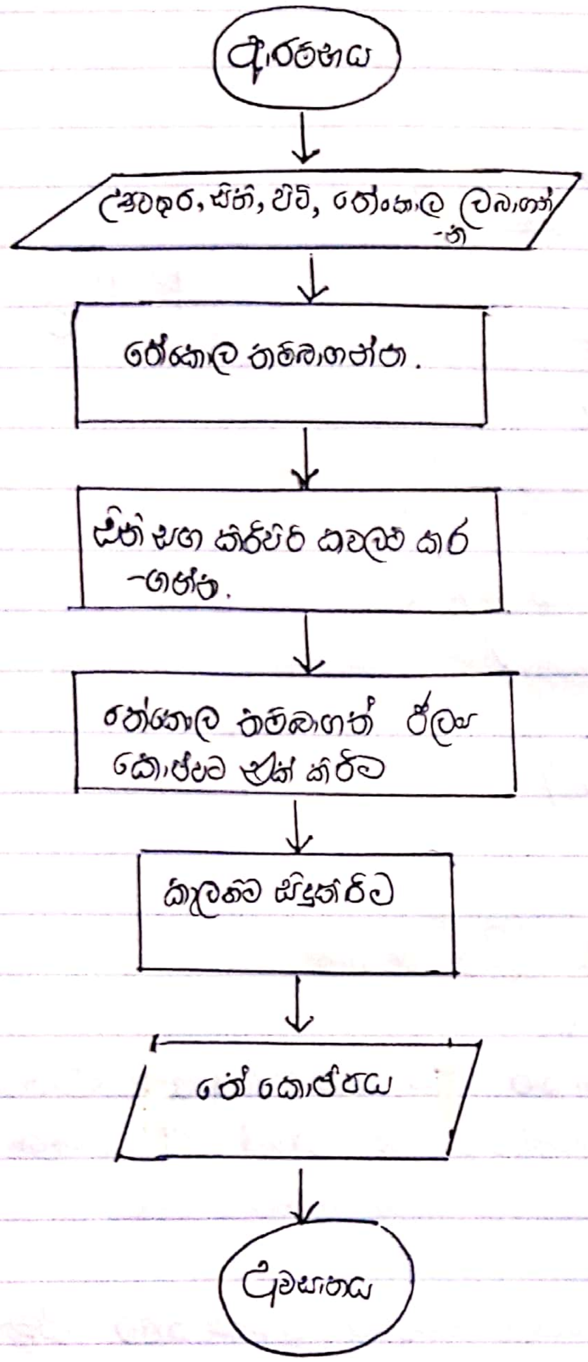
01. සංඛ්‍යා 02ක් ලබාගෙන එහි මධ්‍යන්‍යය (Average) ප්‍රදර්ශනය කිරීම සඳහා, ගැලීම් සටහනක් අඳින්න.



02. තේ එකක් සෑදීම දැක්වෙන ඇල්ගොරිතම සටහනක් අඳින්න.

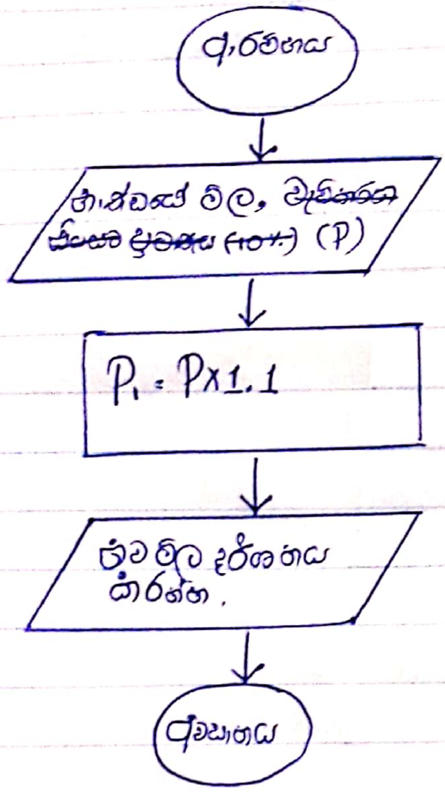


Scanned by CamScanner



ඕනෑම භාණ්ඩයක මිල ලබාගෙන 10% කින් වැඩිකර නව මිල දර්ශනය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.

Scanned by CamScanner

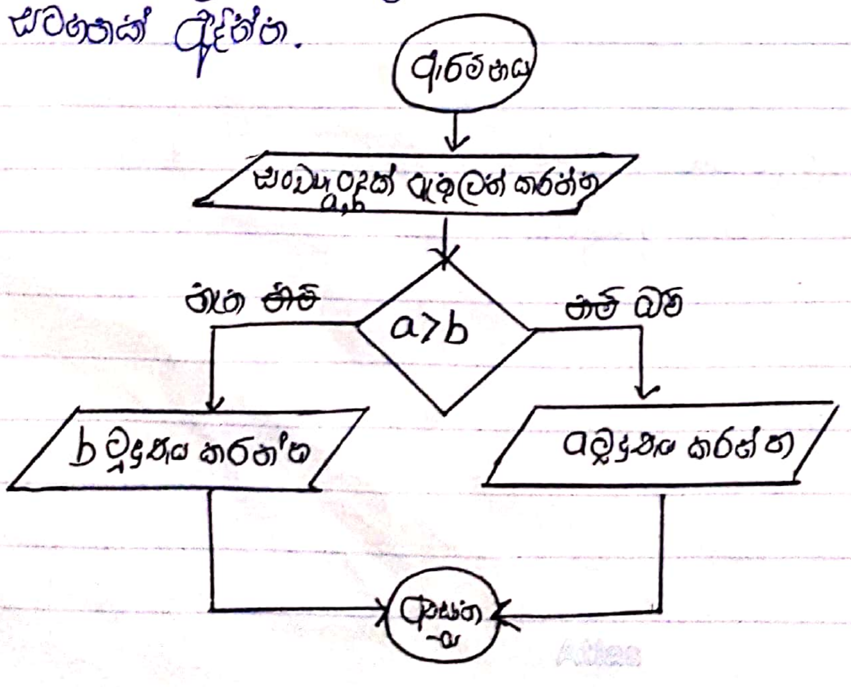


$P = \frac{110}{100} = 1.1$

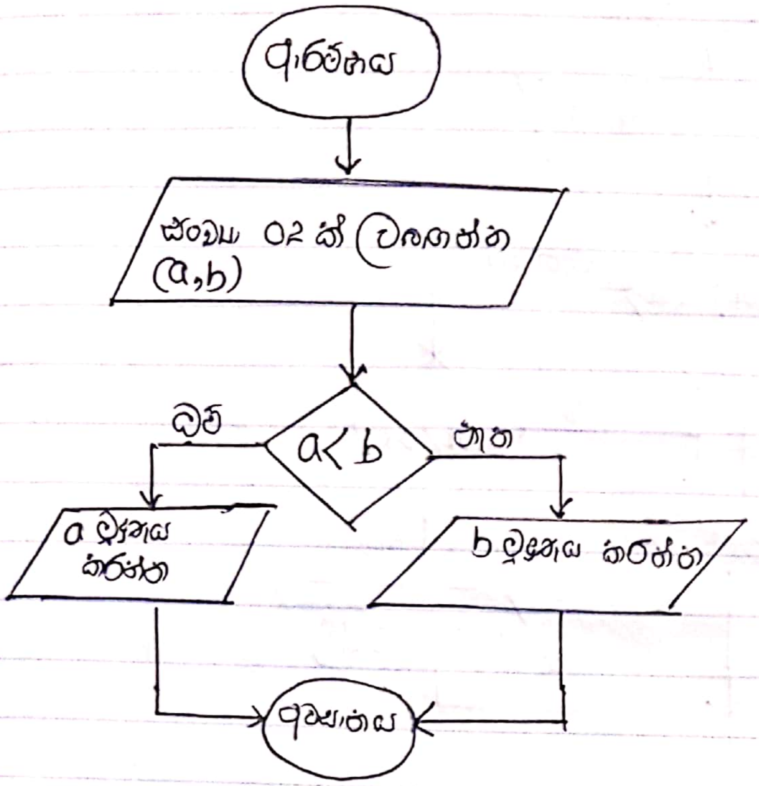
තේරීම් පාලන ව්‍යුහයක් සහිත ගැලීම් සටහන්

කිසියම් කොන්දේසියක් පරීක්ෂා කර එය සත්‍ය වන්නේ නම්, එක් ගමන් මාර්ගයකද, අසත්‍යය වන්නේ නම්, තවත් ගමන් මාර්ගයකද, ගමන් කරලීම සඳහා තේරීම් පාලන ව්‍යුහය භාවිත කරයි.

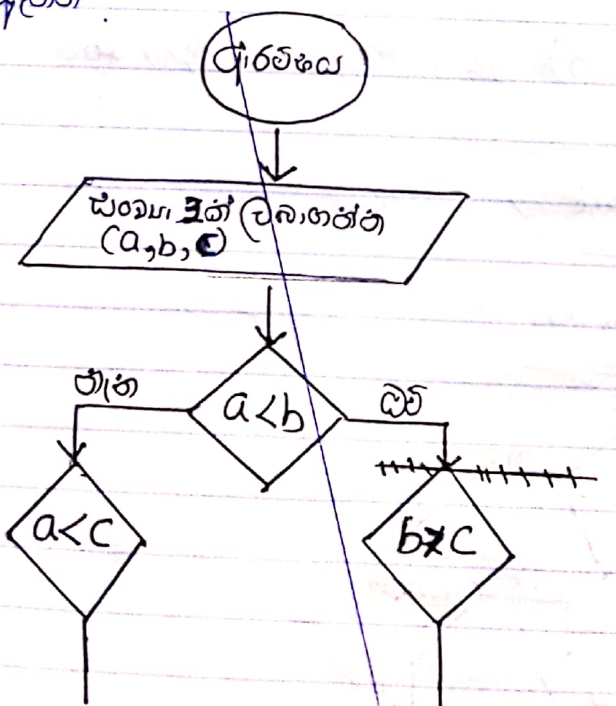
(01). සංඛ්‍යා දෙකක් ලබාගෙන වැඩි විශාල සංඛ්‍යාව මුද්‍රණය කිරීම සඳහා, ගැලීම් සටහනක් අඳින්න.

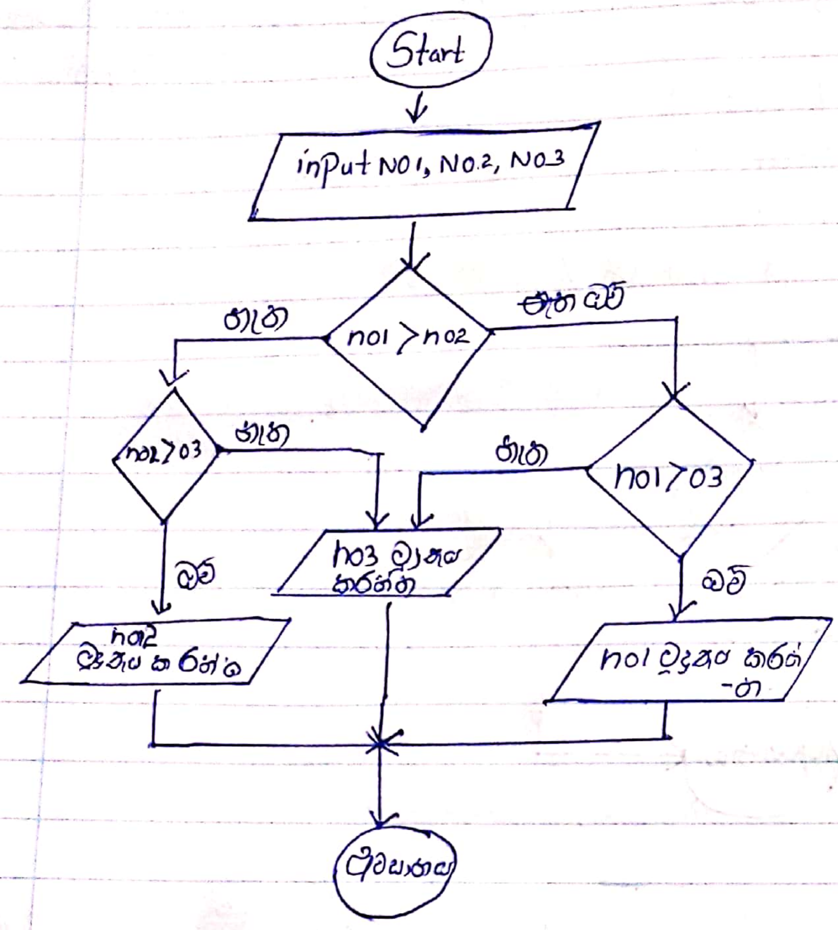


02. සංඛ්‍යා 02ක් ලබාගෙන ඉන් කුඩා සංඛ්‍යාව මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.

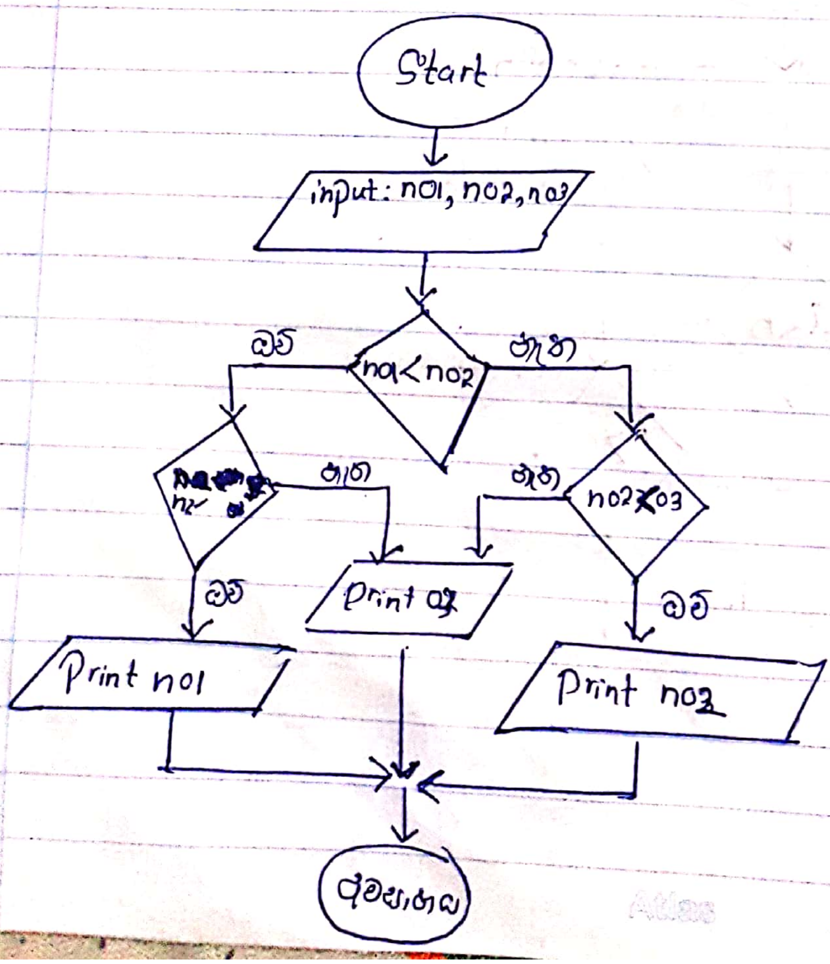


03. සංඛ්‍යා 03ක් ලබාගෙන ඉන් විශාලම සංඛ්‍යාව මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.



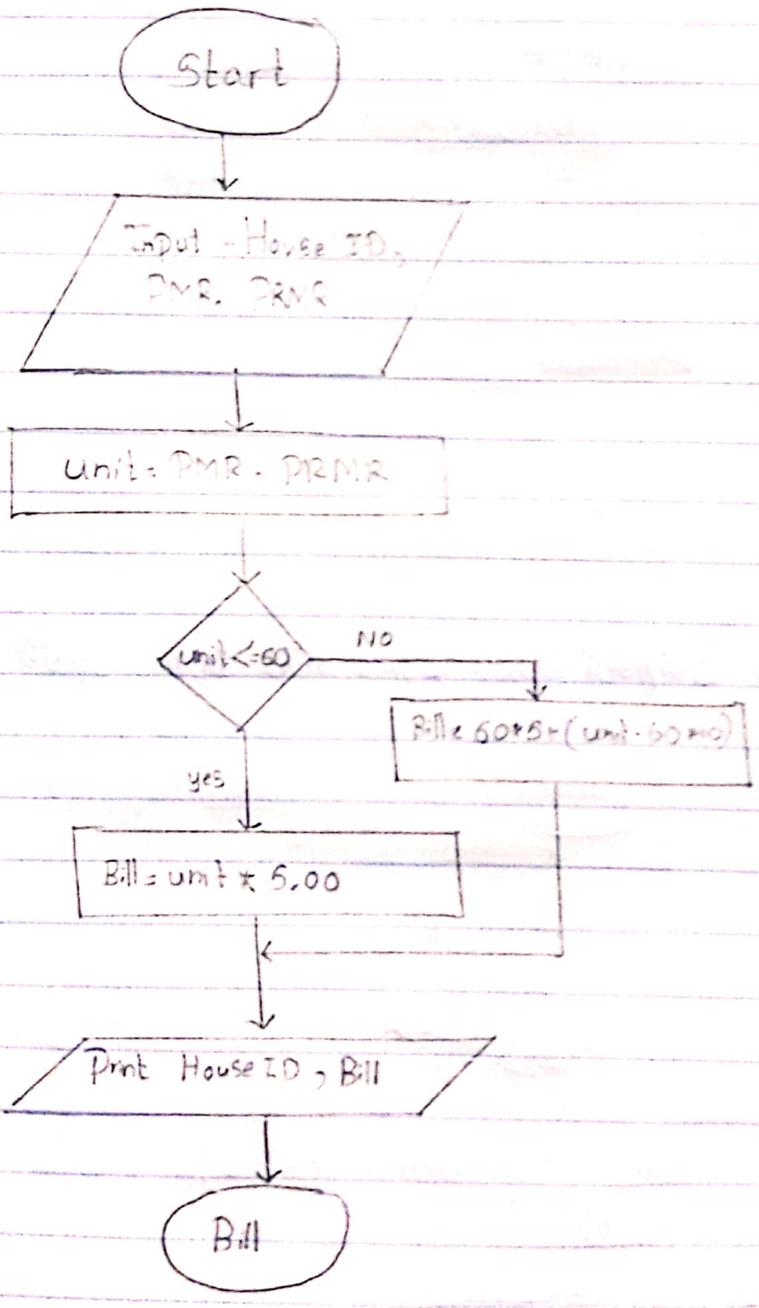


සංඛ්‍යා 03 ක් ලබාගෙන ඉන් කුඩාම සංඛ්‍යාව මුද්‍රණය කිරීම.



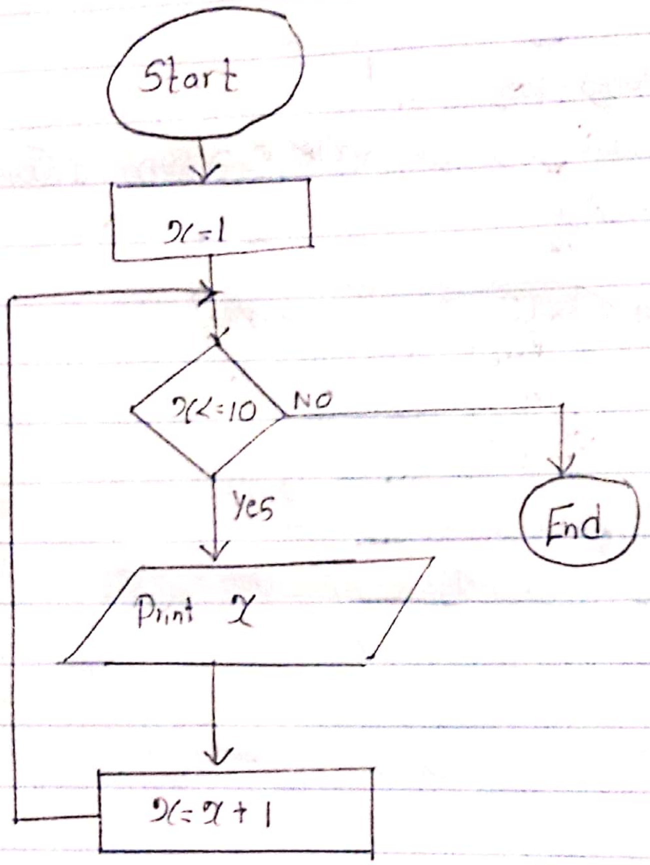
- නිවසක විදුලි බිල ගණනය කිරීම සඳහා මනු කියවන්නා විසින් වර්තමාන කියවීම (PMR) සහ පෙර කියවීම (PRMR) ලියාපදිංචියට ගනු ලබයි. පළමු විදුලි ඒකක 60ක් දක්වා (Unit) සඳහා රු 5 බැගින්ද, 60ට වඩා වැඩිවන සෑම ඒකකයක් සඳහාම රු 10 බැගින්ද අයකරනු ලබයි.

නිවෙස් හිමියාගේ නිවෙස් අංකය (house ID), PMR සහ PRMR ආදාන ලෙස ලබාගෙන නිවෙස් අංකය සමග එහි විදුලි බිල ප්‍රතිදානය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.

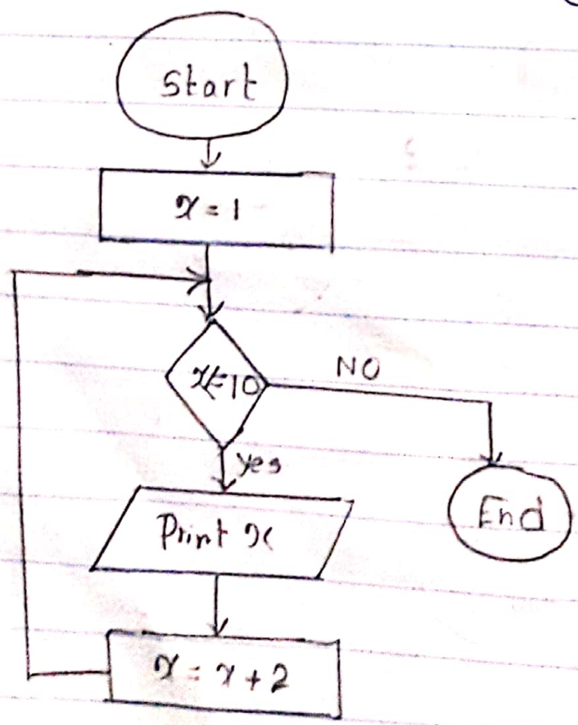


පුනරාවර්තන පාලන ව්‍යුහය සහිත ගැලීම් සටහන්

* 1 සිට 10 දක්වා ගණිත සංඛ්‍යා මුද්‍රණය කිරීමට ගැලීම් සටහනක් අඳින්න.

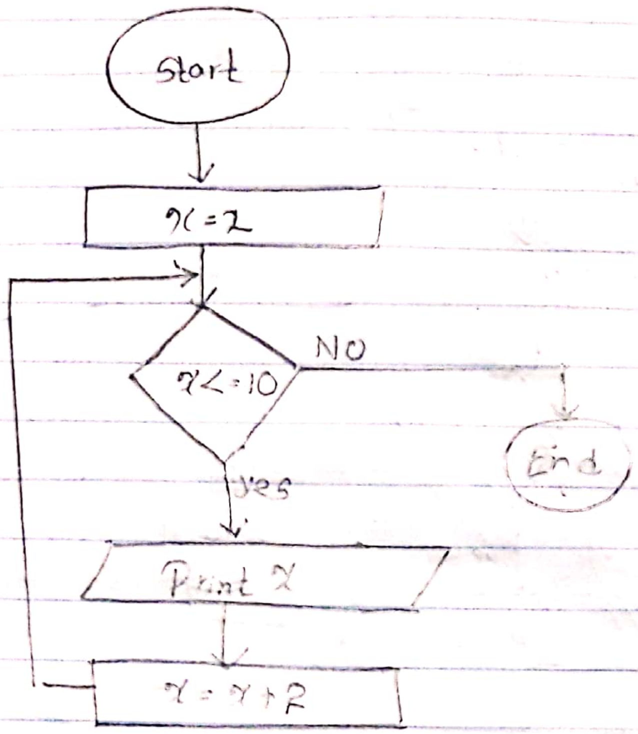


1 සිට 10 දක්වා ඔත්තේ සංඛ්‍යා මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.

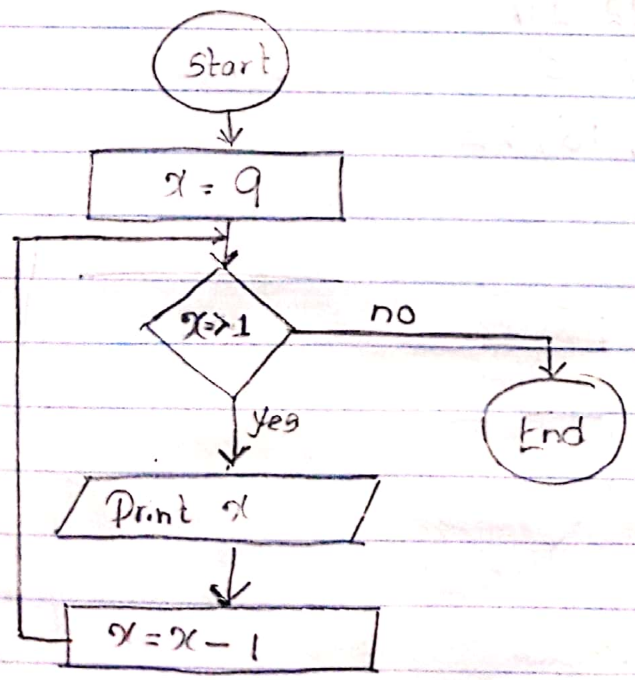


Scanned by CamScanner

1 - 10 දක්වා ඉරට්ටේ සංඛ්‍යා මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න.



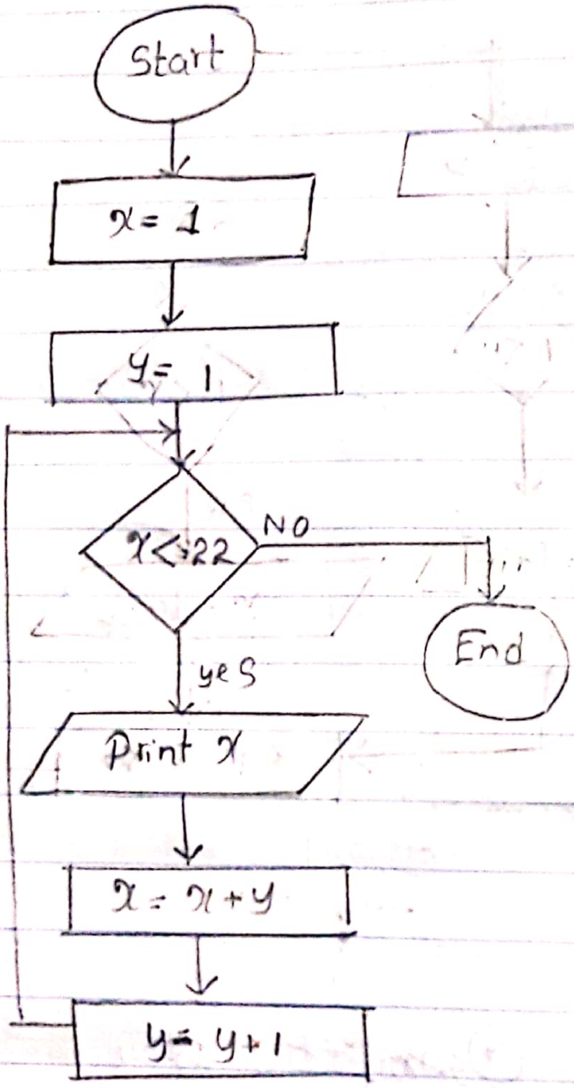
9, 8, 7, 6, 5, 4, 3, 2, 1 මුද්‍රණය කිරීම සඳහා



No :

Date :

1, 2, 4, 7, 11, 16, 22



x : 2 4 7 11 16 22 24

y 2 3 4 5 6 7 8

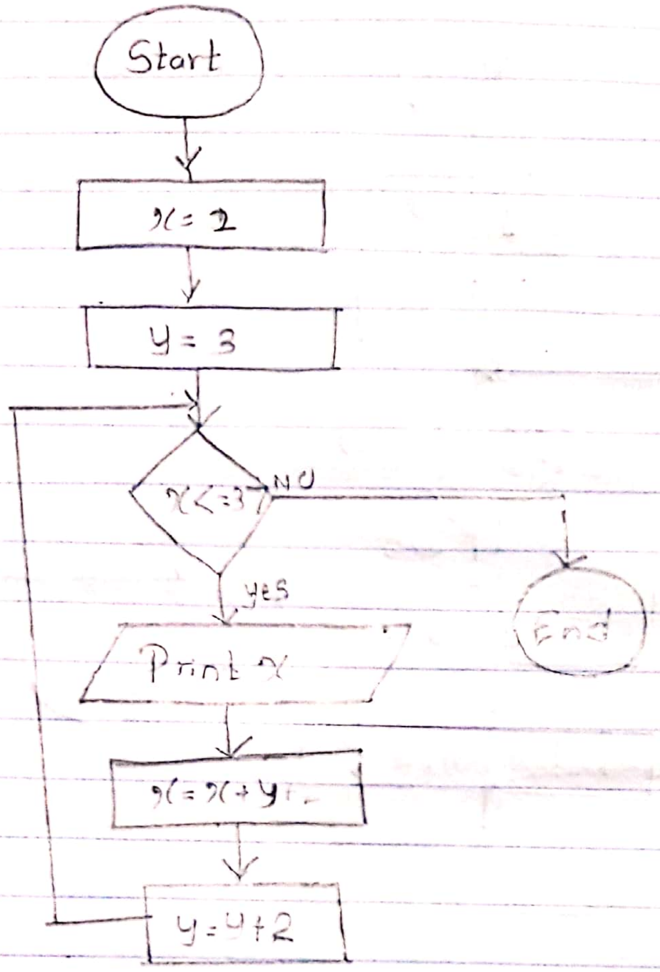
1 2 4 7 11, 16, 22

Scanned by CamScanner

No. Date:

2, 5, 10, 17, 26, 37

3 5 7 9 11



1, 1, 2, 3, 5, 8, 13, 21, 34, 55

0 1 1 2 3 5 8 13 21

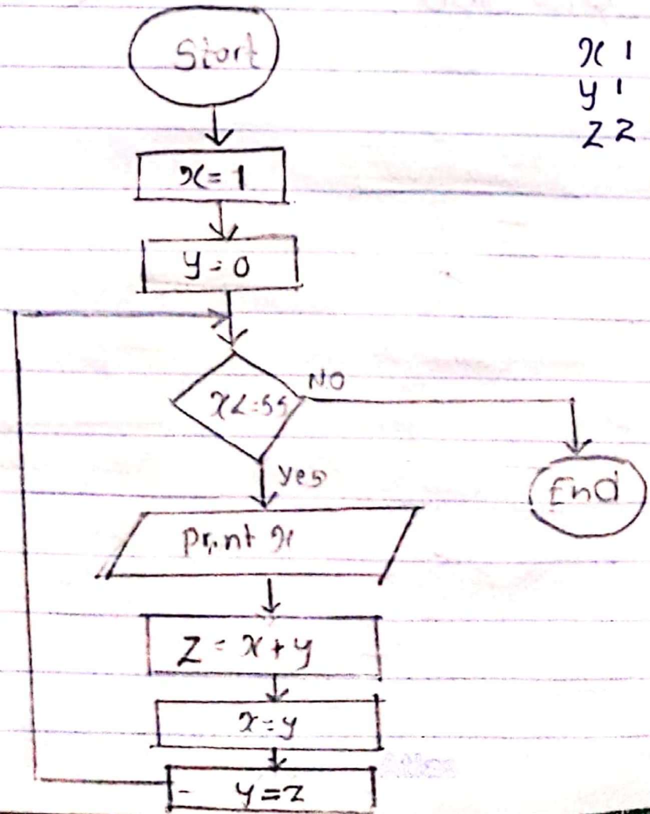
x y z

Z = x + y

x = y

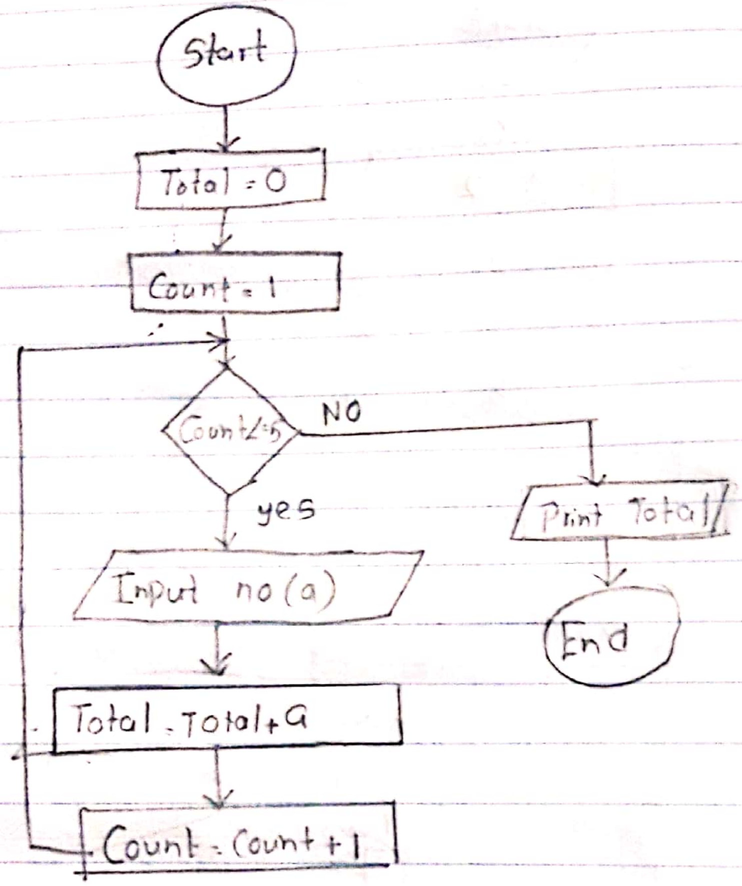
y = Z

x	1	1	2	3
y	1	2	3	8
Z	2	3	5	13

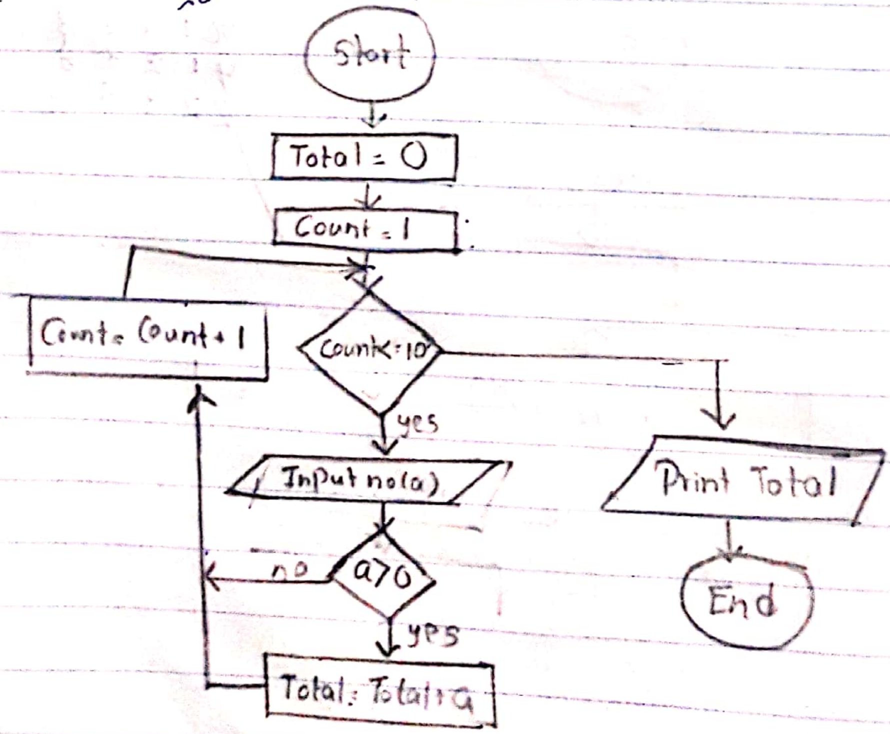


Scanned by CamScanner

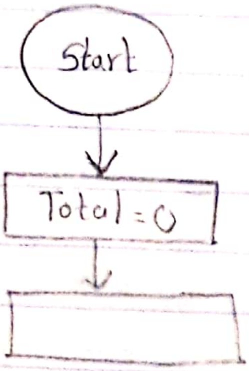
ධන පූර්ණ සංඛ්‍යා: 05 ක් වෙන්වෙන් වශයෙන් ඇතුලත් කර ඒවා සම්පූර්ණ එකතුව මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න!



පරිශීලකයාගෙන් ධන සංඛ්‍යා 10ක් වෙන්වෙන්ව ලබාගෙන ඒවායින් + සංඛ්‍යා එකතුව පමණක් මුද්‍රණය කිරීම.

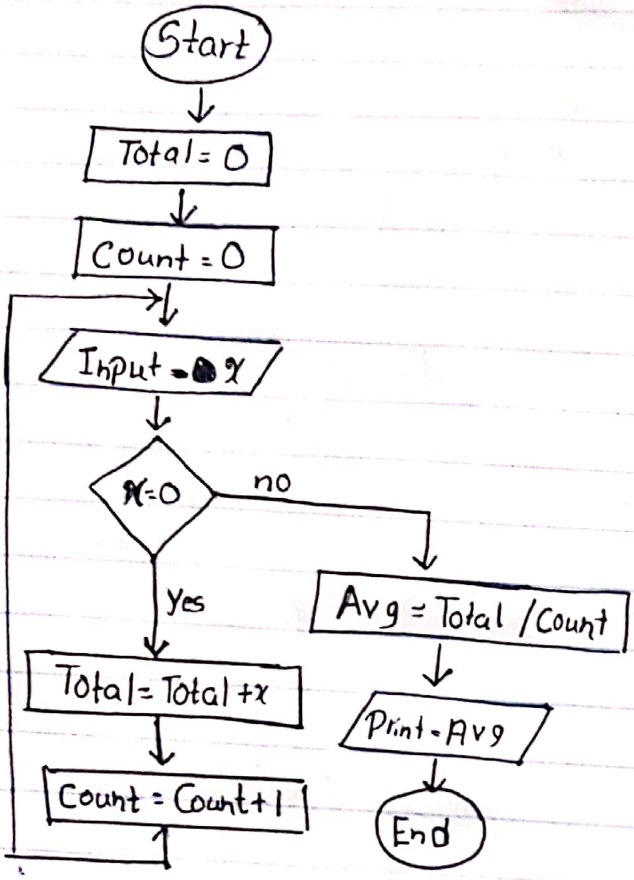


සංඛ්‍යා 10 ක් ඇතුළත් කර එහි ඔත්තේ සංඛ්‍යාවල එකතුව සහ ඉරට්ටේ සංඛ්‍යාවල එකතුව ද වෙන වෙනම මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් යොදන්න.



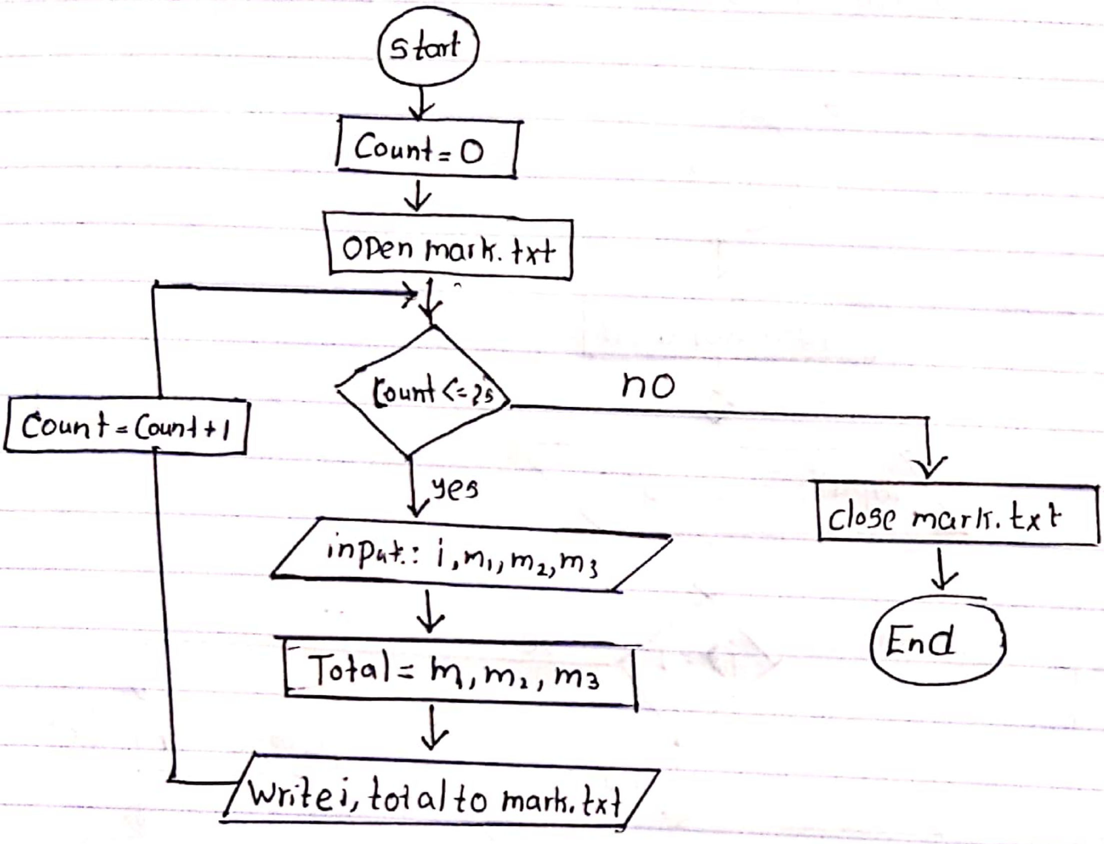
සංඛ්‍යා සමූහයක් ලබාගෙන ඒවායේ මුළු එකතුව මුද්‍රණය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න. එහිදී පරිශීලකයා, ඇතුළත් කළවිට ලූපය නතරවේ ඍණාත්මක ලබාදිය යුතුය.

සංඛ්‍යා කිහිපයක් ලබාගෙන එහි මධ්‍යන්‍යය ගණනය කිරීම සඳහා ගැලීම් සටහනක් අඳින්න. එහිදී 0 ඇතුලත් කළ විට ගණනය නතර වී මධ්‍යන්‍යය ගණනය විය යුතුය.



පරිශීලකයාගෙන් කිසියම් සංඛ්‍යාවක් ලබාගෙන එහි ක්‍රමාරෝපිතය සෙවීම සඳහා ගැලීම් සටහනක් අඳින්න.

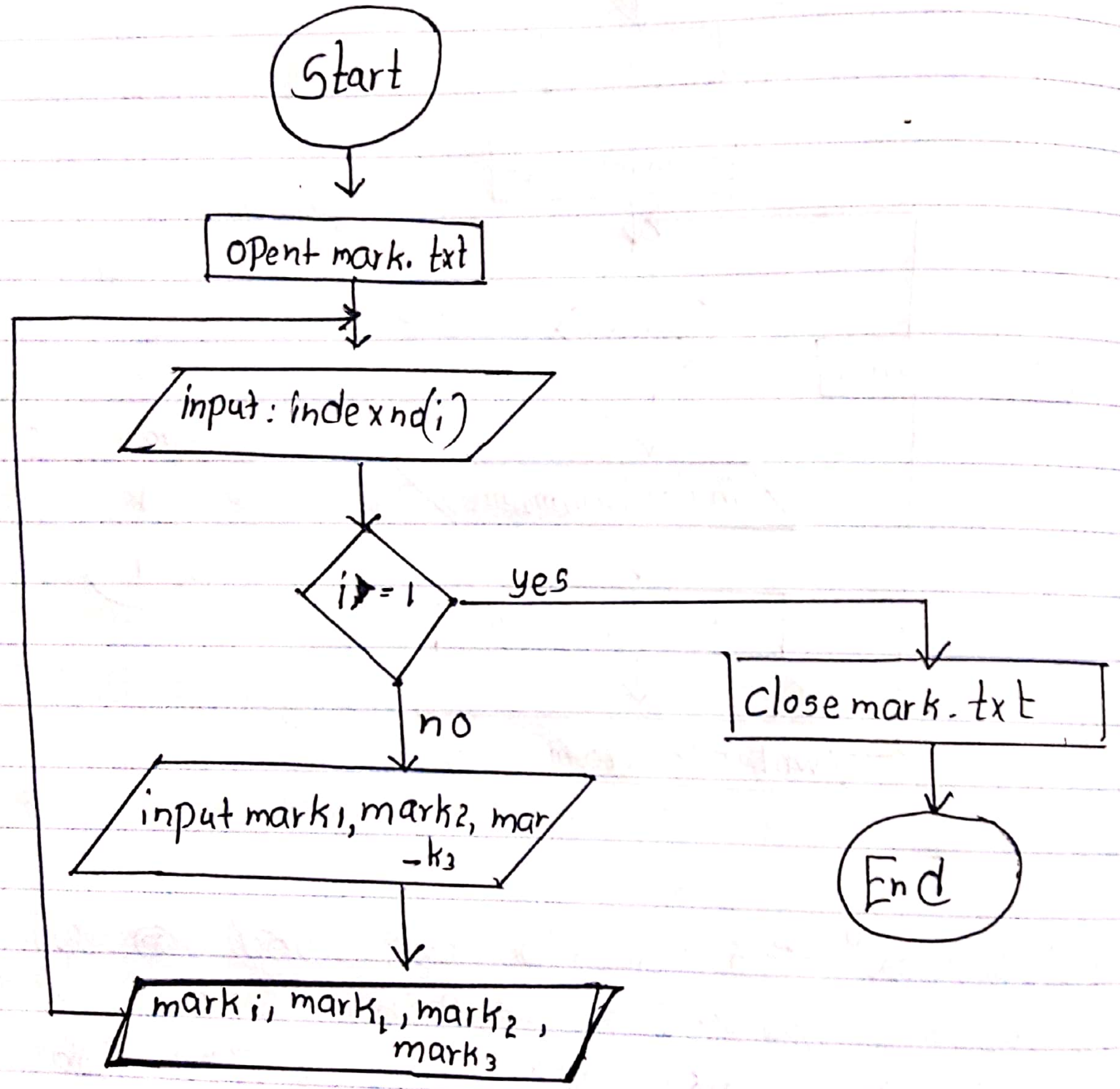
සිසුන් 25 දෙනෙකු සිටින පන්තියක සිසුවාගේ විභාග අංකය (i) සහ විෂයන් 03 ක ලකුණු (m_1, m_2, m_3) ලබාගෙන සිසුවාගේ විභාග අංකය සහ ලකුණුවල එකතුව mark.txt නම් ගොනුවක ලිවීම සඳහා ගැලීම් සටහනක් අඳින්න.



වාර විභාගයකදී ළමුන් ලබාගන්නා ලකුණු වාර්තා කිරීම සඳහා පරිගණක ක්‍රමලේඛයක් ලියන ලෙස ඔබගේ ගුරුතුමා විසින් ඔබගෙන් ඉල්ලීමක් කර තිබේ. සෑම සිසුවෙක්ම එකම ප්‍රශ්න පත්‍ර 3 කට පෙනී සිටි ඇති අතර සෑම ලකුණක්ම දී ඇත්තේ 100 වන අතර එය නිඛිල අගයක් වේ. සෑම සිසුවෙකුටම අනන්‍ය සුචි අංකයක් මගින් හඳුනා ගන්නා අතර සුචි අංකයද නිඛිලයක් වේ. ඔබ විසින් පහත පෙන්වා දී ඇති ආකාරයට mark.txt නමැති පාඨ ගොනුවේ වාර්තා කළ යුතුවේ. මෙහි index no x: X යනු x වැනි සිසුවාගේ සිසු අංකය, mark x, y x වන සිසුවා, y ප්‍රශ්න පත්‍රයට ගත් ලකුණු සංඛ්‍යාවයි.

වරකට එක අයිතමය බැගින් යතුරු පුවරුව මගින් සිසුන්ගේ සූචි අංකය හා ලකුණු ඇතුලත් කල යුතුවේ. සූචි අංකය -1 ලෙස ඇතුලත් කලවිට ක්‍රමලේඛය නැවතිය යුතුවේ.

ගැලීම සටහනක් භාවිතයෙන් ඉහත ක්‍රමලේඛයට ඇල්ගොරිතමයක් ඇඳ



ව්‍යාජ කේත

ඇල්ගොරිතමයක් වචන ආකාරයෙන් ඉදිරිපත් කිරීම සඳහා ව්‍යාජකේත භාවිතා කරයි.

1. අනුක්‍රම පාලන ව්‍යුහයක් සහිත පාලන ව්‍යුහ

(i). සංඛ්‍යා 2ක් ඇතුලත් කර එහි එකතුව දර්ශනය කිරීම සඳහා ව්‍යාජ කේතයක් ලියන්න.

Begin	ආරම්භය
Input no_1 and no_2	පළමු එකතු සංඛ්‍යා, 2 වන එකතු සංඛ්‍යා, ට
Total = $no_1 + no_2$	එකතුව= පළමු සංඛ්‍යාව + දෙවන සංඛ්‍යාව
Display Total	එකතුව දර්ශනය කරන්න
End	අවසානය

(ii) සංඛ්‍යා 2ක් මධ්‍යන්‍යයේ එහි මධ්‍යන්‍යය දර්ශනය කිරීමට ව්‍යාජකේතයක් ලියන්න.

Begin

Input no_1, and no_2

$$Avg = \frac{no_1 + no_2}{2}$$

Display Avg.

End

Scanned by CamScanner

② තේරීම් පාලන ව්‍යුහයක් සහිත ව්‍යාජකේත

If, Then Else

කිසියම් සංඛ්‍යා 2ක් ලබාගෙන ඉන් විශාලතම සංඛ්‍යාව මුද්‍රණය කිරීම සඳහා ව්‍යාජකේතයක් ලියන්න .

```
Begin
 Input no1 and no2
 If (no1 > no2) then
Print no1
Else .
 Print no2
End If
End
```

සිසුවෙකු විභාගයකදී ගත් ලකුණු mark ලෙස ලබාගෙන එය 50 වැඩිහෝ සමාන නම්, Pass ලෙසද නැතිනම් fale ලෙසද මුද්‍රණය කිරීම සඳහා ව්‍යාජකේතයක් ලියන්න .

```
Begin
 Input mark
 If (mark >= 50) then
Print "Pass"
 Else
Print "fale"
 End If
  End
```

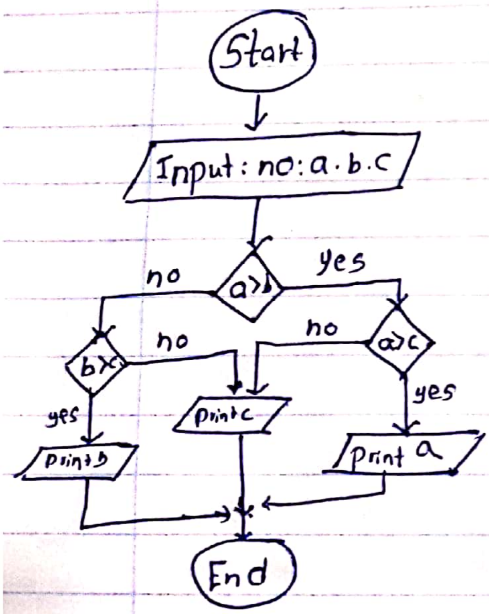
Scanned by CamScanner

- ශිෂ්‍යයෙකු විභාගයකදී ගත් ලකුණු ලබාගෙන එය 70 >= නම A ලෙසද 50 >= B ලෙසද 40 >= නම් S ලෙසද 40ට අඩු නම F ලෙසද දර්ශනය කිරීමට ව්‍යාජකේතයක් ලියන්න.

```
Begin
 Input mark
 If (mark>=70) then
  Print A
 ~~If~~ Else If (mark>= 50) then
  Print B
 Else If (mark >= 40) then
  Print S
  Else Print F
 End IF
 End If
 End If
  End
```

Scanned by CamScanner

* සංඛ්‍යා 3ක් ලබාගෙන ඉන් විශාලතම සංඛ්‍යාව මුද්‍රණය කිරීම සඳහා ව්‍යාජකේතයක් ලියන්න.




```
Begin
   input number 1 as a
   input number 2 as b
   input number 3 as c
   If a>b Then
       If a>c Then
           Print a
       Else
           Print c
       End If
   Else If b>c Then
           Print b
       Else
           Print c
       End If
   End If
End.
```

පුනරාවර්තන සහිත ව්‍යාජ කේත

for next පුනරාවර්තනය...

* පුනරාවර්තනය එක වාරගණන දන්නා අවස්ථාවලදී for next භාවිත කල හැකිය. එනම්, වාරගණන නොදන්නා අවස්ථාවලදී මෙය භාවිත කළ නොහැකිය.

Scanned by CamScanner

(01). 1 සිට 10 දක්වා ගණිත සංඛ්‍යා මුද්‍රණය කිරීම සඳහා ව්‍යාජකේතයක් ලියන්න.

```
Begin
for i = 1 to 10
    Print i
Next i
End
```

(02). 1 සිට 10 දක්වා, Ø ඔත්තේ සංඛ්‍යා මුද්‍රණය කිරීම සඳහා, ව්‍යාජකේතයක් ලියන්න.

i++ (දෙකක් +)

```
Begin
For i = 1 to 10 i++
    Print i
Next i
End
```

(03) සංඛ්‍යා 10ක් ලබාගෙන එහි මුළු එකතුව මුද්‍රණය කිරීම සඳහා ව්‍යාජකේතයක් ලියන්න (For භාවිතයෙන්)

```
Begin
tot = 0
for i = 1 to 10
    input no as a
    total = total+a
Next i
Print tot
End
```

Scanned by CamScanner

(04). සංඛ්‍යා 10ක් ලබාගෙන එහි ඔත්තේ සංඛ්‍යාවල එකතුව ලබාගෙන මුද්‍රණය කිරීම සඳහා for loop එකක් ලියන්න.

```
Begin
 For i = 1 to 10 i++
   Print i
 Next i
```

```
Begin
otot = 0
  for i = 1 to 10
    input no as a
    If a%2 == 1 Then
      otot = otot + a
    End If
  Next i
Print otot
End
```

While Do පුනරාවර්තනය

පුනරාවර්තනය වන

* වාර ගණන දන්නා හෝ නොදන්නා ඕනෑම අවස්ථාවක While do පුනරාවර්තනය යොදා ගත හැකිය.

(01). 1 සිට 10 දක්වා මුද්‍රණය කිරීම සඳහා While do පුනරාවර්තනය භාවිතයෙන් ව්‍යාජකේතයක් ලියන්න.

```
Begin
 i = 1
 While i<=10 Do
    Print i
     i = i+1
  End While
 End
```

Scanned by CamScanner

(02). සංඛ්‍යා 10ක් ඇතුලත් කර එහි සම්පූර්ණ එකතුව ලබාගැනීම සඳහා, while ක්‍රමලේඛයෙන් ලියන්න.

```
Begin
tot = 0
While i = 1 to 10 Do
  input no as a
  tot = tot + a
  -
```

```
Begin
 i = 1
 tot = 0
 while  i <= 10 Do
   input no as a
   tot = tot + a
   i = i + 1
 End while
 Print tot
End
```

(03). සංඛ්‍යා සමූහයක් ලබාගෙන ඒවායේ ඉරට්ටේ සංඛ්‍යාවල එකතුව පමණක් මුද්‍රණය කිරීම සඳහා While ක්‍රමලේඛයක් ලියන්න. මෙහිදී 0 ඇතුලත් කළහොත් ලූපය නතර වී ඉරට්ටේ සංඛ්‍යා වල එකතුව මුද්‍රණය විය යුතුය.

```
Begin
ETotal = 0
 Input number as a
  If a % 2 = 0 Then
   Etot = Etot + a
  End while
   Print Etot
  End
```

```
Begin
 etotal = 0
 input no as a
 While a =! 0 Do
   If a % 2 = 0 Then
     etot = etot + a
   End if
   Input no as a
 End while
 Print etot
 End
```

(01) ඉත ගණනෙහිම ඔත්තේ සංඛ්‍යා ගණන හා ඉරට්ට සංඛ්‍යා ගණන මුද්‍රණය කිරීමට While Do ආකාරයෙන් ව්‍යාජ කේතය ලියන්න.

```
Begin
 Ocount = 0
 ecount = 0
 etot = 0
 input no as a
 while a=!0 Do
   If a%2 = 0 Then
       ecount = ecount + 1
       etot   = etot + a
   Else
      ocount = ocount + 1
   End if
   input number as a
  End while
  Print etot
  Print " o tot number" ocount
  Print "etot number" ecount
 End.
```

Repeat Until පුනරාවර්තනය

Repeat Until පුනරාවර්තනය පුනරාවර්තනය වන වාරගණන දන්නා අවස්ථාවලදී මෙන්ම නොදන්නා අවස්ථාවලදීද භාවිත කල හැකිය. මෙහි ඇති විශේෂත්වය වන්නේ කොන්දේසිය පරීක්ෂා ~~කිරීමේ~~ කිරීම සිදුවන්නේ පුනරාවර්තනයෙන් පසු එය පසු කේතයකින් ලියුව විමසීම. එසේම මෙහිදී කොන්දේසිය අසත්‍යවන තුරු වරක් පුනරාවර්තනය සිදුවේ.

01. සංඛ්‍යා සමූහයක් ලබාගෙන එහි මුළු එකතුව මුද්‍රණය කිරීම සඳහා Repeat Until පුනරාවර්තනය භාවිතයෙන් ඇල්ගොරිතමයක් ලියන්න. මෙහිදී 0 ඇතුලත් කරන විට ලූපය නතරවෙ.

```
Begin
 tot = 0
 Repeat
   input no as a
    tot = tot + a
  Until a=0
  Print tot
End
```

18/12/2018

පරිගණක භාෂාවන්ගේ පරිණාමය

පරිගණක භාෂා පරිණාමය පරම්පරා 4 ක් ඔස්සේ සාකච්ඡා කළ හැකිය.

01. පළමු පරම්පරාවේ පරිගණක භාෂා (Machine language)
02. දෙවන පරම්පරාවේ පරිගණක භාෂා (Assembly language)
03. තුන්වන පරම්පරාවේ පරිගණක භාෂා (High level language)
04. හතරවන පරම්පරාවේ පරිගණක භාෂා (Artificial language/කෘත්‍රිම භාෂා)

01. යාන්ත්‍රික භාෂාව (Machine language)

* උපදෙස් ඇතුලත් කිරීම සඳහා 1,0 යොදාගන්නා ලදී.
* ක්‍රමලේඛකයාට ද්වීසංඛ්‍යා පිළිබඳව හොඳ අවබෝධයක් තිබිය යුතුය.
* යන්ත්‍රය මත යැපෙන භාෂාවකි.
* උපදෙස් ක්‍රියාත්මක වීම ඉතා අධි වේගවත්වේ.

02 ඇසෙම්බ්ලි භාෂාව (සංකේතාත්මක භාෂා)

* උපදෙස් ඇතුලත් කිරීම සඳහා 1,0 වෙනුවට සංකේත භාවිතවිය.
* භාෂා පරිවර්තක මෘදුකාංගයක් අවශ්‍යයවේ. ඒසඳහා ඇසෙම්බල (Assembler) නම් අර්ථ විනයනය යොදාගෙන ඇත.
* යන්ත්‍රය මත යැපෙන භාෂාවකි.
* උපදෙස් ක්‍රියාත්මක වීම වේගය යාන්ත්‍රික භාෂාවට සාපේක්ෂව අඩුවේ.

03. තුන්වන පරම්පරාවේ පරිගණක භාෂා (උසස් මට්ටමේ පරිගණක භාෂා)

* උපදෙස් ඇතුලත් කිරීම සඳහා සරල ඉංග්‍රීසි භාවිත කරනු ලබයි.
* ක්‍රමලේඛනය කිරීම ඉතා පහසුය.
* යන්ත්‍රය මත රඳාපවතින භාෂාවක් නොවේ.
* යාන්ත්‍රික භාෂාව බවට පරිවර්තනය කළ යුතුය. ඒ සඳහා අර්ථවිනයාස හෝ සම්පාදක යොදාගනු ලබයි.

උදා: ෆෝට්‍රන්, Cobol, C, pascal, Python

04. හතරවන පරම්පරාවේ පරිගණක භාෂා (කෘතිම පරිගණක භාෂා)

* සිතාමතා ක්‍රියාත්මක කළහැකි ~~මිනිස්~~ මිනිසුන්ගේ ක්‍රියාකාරකම්, හැසිරීම් වැනි, අනුබද්ධ කළහැකි උපකරණ නිර්මාණයට යොදාගැනේ.

උදා: lisp, Prolog

* රොබෝ යන්ත්‍ර නිපදවීමේදී මේවා යොදාගැනේ.

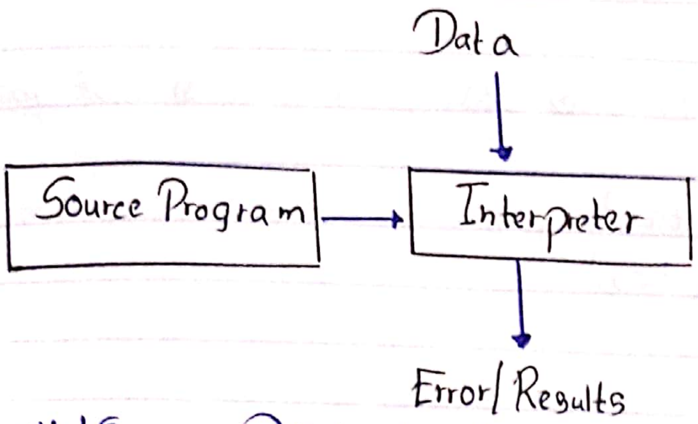
◄ භාෂා පරිවර්තක මෘදුකාංග

* ඉහළ මට්ටමේ පරිගණක භාෂාවලින් ලියනු ලබන ක්‍රමලේඛ පරිගණක සකසනයට තේරුම්ගත නොහැක. එනිසා ඒවා සකසනයට තේරුම් ගත හැකිවන පරිදි යාන්ත්‍රික භාෂාව බවට පරිවර්තනය කළයුතුය.
* මෙසේ පරිවර්තනය කිරීම සඳහා යොදාගනු ලබන මෘදුකාංග භාෂා පරිවර්තක මෘදුකාංග ලෙස හඳුන්වයි.
* මෙය ප්‍රධාන ආකාර 02කි.
    * අර්ථ විනයාස (Interpiter)
    * සම්පාදක (Compiler)

• අර්ථ විනයාසය

* අර්ථ විනයාසය මගින් ක්‍රමලේඛයෙහි පියවරෙන් පියවර යාන්ත්‍රික භාෂාව බවට පරිවර්තනය කිරීම සිදුකරයි.
* Phython, Visual Basic වැනි භාෂාවල අර්ථ විනයාස භාවිතාවේ.
* මෙහිදී මෘදුකාංගය ධාවනය කලයුතු සෑම අවස්ථාවකදීම පරිවර්තනය කළයුතුය.

Scanned by CamScanner



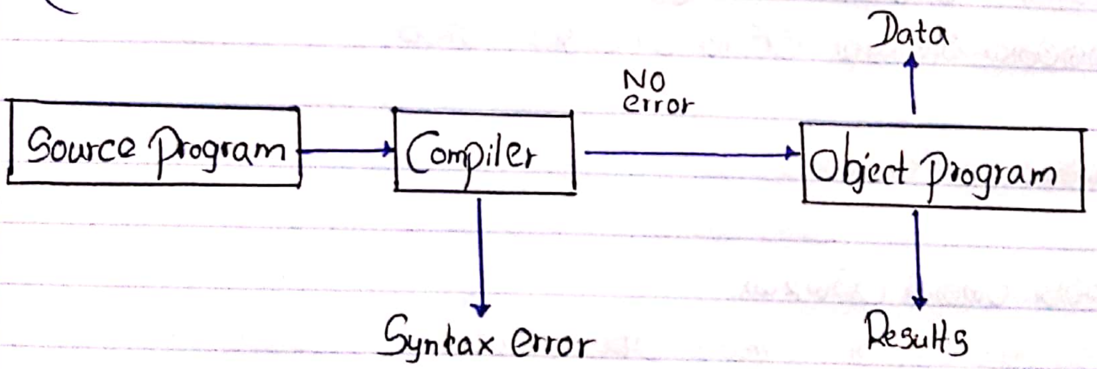
අර්ථ වින්‍යාසක මගින් ක්‍රමලේඛ ක්‍රියාත්මක කිරීම.

- සම්පාදක

*සම්පාදක වලදී සම්පූර්ණ ක්‍රමලේඛනයම එකවරම යාන්ත්‍රික භාෂාව බවට පරිවර්තනය වීම සිදුවේ.

* එහිදී වෙනම ගොනුවක් නිර්මාණය වන අතර, එමගින් නැවත නැවත Run කිරීමෙන් පරිවර්තනය කිරීම අවශ්‍ය නොවේ.

(උදා: Pascal



Python ක්‍රමලේඛ භාෂාව

* Python යනු, උසස් මට්ටමේ ක්‍රමලේඛ භාෂාවක් වන අතර මෙම භාෂාව 80 දශකයේ අගභාගයේ සහ 90 දශකයේ මුල් භාගයේදී Guido Van Rossum විසින් නිර්මාණය කරන ලදී.

* Python භාෂාව,
    - අර්ථ වින්‍යාසක භාෂාවකි.
    - වස්තු නැඹුරු භාෂාවකි.
    - ව්‍යුහගත භාෂාවකි.
    - සරල මෙන්ම ප්‍රබල භාෂාවකි. (library file)
    - library file විශාල ප්‍රමාණයක එකතුවක් ඇත.

Scanned by CamScanner

පරිගණක භාෂා නිර්වචනය කිරීම.

පරිගණක භාෂා නිර්වචනය කිරීම සඳහා ප්‍රධාන සංඝටක 2 ක් අවශ්‍ය වේ.
ඒවා නම්,

- කාරක රීති (Syntax)
- ශබ්දාර්ථ (Semantics)

* කාරක රීති (Syntax)

- කාරක රීති මගින් භාෂාවේ ඇති ව්‍යාකරණ නීති අනුව සංකේත සම්බන්ධ කළ යුතු ආකාරය නිර්වචනය කරයි. එමගින් සංකේත වල ව්‍යුහයන්ගේ නිවැරදිභාවය පිළිබඳ පමණක් ~~අන්තර~~ සැලකිල්ලමත් වන අතර අර්ථය පිළිබඳව සැලකිල්ලක් නොදක්වයි.

* ශබ්දාර්ථ (Semantics)

මෙමගින් භාෂාවක ඇති ව්‍යාකරණකුල ව්‍යුහයන්ට ඇතුළත් අර්ථ පවරා එමගින් භාෂාවකින් සම්පාදනය කළ වැඩසටහනක් ක්‍රියාත්මක කිරීමේදී අනුගමනය කළ යුතු චර්යාව නිර්වචනය කරයි.

පයිතන් වගන්ති වර්ග

පයිතන් වගන්ති වර්ග 2 කි.

01. සරල වගන්ති (Simple Statement)
02. සංයුක්ත වගන්ති (Compound Statement)

01. සරල වගන්ති...

* එක් වගන්තියක් පමණක් ඇති පයිතන් වගන්තියක් සරල වගන්තියක් වේ.

* x=5

* Print(x)

* එක්පේළියක ";" Semicolon මගින් වෙන්වූ වගන්ති කිහිපයක් ලිවිය ~~02. සංයුක්ත~~ හැක. * x=5; y="Saman"

02 සංයුක්ත වගන්ති....

* පයිතන් වගන්ති එකක් හෝ කිහිපයක් සංයුක්ත වගන්තියකට ඇතුලත් විය හැකිය.
* එහි ශීර්ෂ (header) සහ එයට අයත් අනුවගන්ති වලින් සමන්විතවේ.
* ශීර්ෂකොටස මුල පදයකින් ආරම්භ වී කෝලන් (";") අවසන්වේ
* එයට අදාළ වගන්ති (Suite) අනුච්ඡේදනය කොට (indent) ලියනු ලැබේ.

උදා:

```
if x=100;
    Print(Excellent)
    y=3
elif x>50;
    Print(Good)
    y=2
elif x>30;
    Print (Must improve)
    y=1
else
    Print (fale)
    y=0
```

ව්‍යක්ත පේළි සම්බන්ධය (Explicit line joining)

පයිතන් කේතයක් පේළියක් "\" අවසන් වේ නම්, එය පසු පේළියත් සමග සම්බන්ධවේ. එනම් එමගින් තනි පයිතන් කේත පේළියක් ලැබෙයි.

උදා:

```
x=1+\
    2
```

ඉහත කේතයෙහි අදහස x=1+2

Scanned by CamScanner

- **Implicit line joining**

"( )", "[ ]", "{ }" මෙම වරහන් තුළ ඇති කේත එකම පේළියකට ලියවෙයි.

උදා:
```
x = {8:"a", 9:
     "b", 10:"c"}
```

මෙන් දැකය `x = {8:'a', 9:'b', 10:'c'}`

Comment in Python

පයිතන් තුළ විවරණ ඇතුලත් කිරීම සඳහා # සලකුණ භාවිතා කරයි.

උදා:
```
# This is python Comment
# Adding two numbers
```

a,b=5,3 # මෙහිදී a=5 ද, b=3 ලෙස ගබඩා කරගනියි.

* පේළි කිහිපයක Comment ඇතුලත් කිරීමට # යොදා ගනියි. එසේම වූ පේළි කිහිපයක Comment ඇතුලත් කිරීමට සඳහා """ """ හෝ ''' ''' යොදා ගනියි.

පයිතන් මූල පද (Key words / Reserved words)

එනම පරිගණක භාෂාවක එම භාෂාවම වෙන්වූ මූලපද වචනයි.

False	try	not	import
Class	True	with	Pass
finally	def	as	break
is	from	elif	except
return	nonlocal	if	in
None	while	or	raise
Continue	and	yield	
for	del	assert	
lambda	global	else	

(False, None, True මුල් අකුරු Capital වේ)

Scanned by CamScanner

දත්ත සහ වස්තු (Data and Objects)

* පයිතන් භාෂාවේ දත්ත වස්තු ලෙස නිරූපණය කරයි.
* සෑම වස්තුවකටම අනන්‍යතාවයක් (identity) වර්ගයක් (Type) සහ අගයක් (Value) ඇත.

* id() - වස්තුවේ ශුද්ධනම ලබාදේ.
* type() - වස්තුවේ වර්ගය ලබාදේ.

```
id(a)
505910704
b=10
id(b)
505910784
type(a)
class 'int'>
type(b)
class 'int'>
C=5.5
id(c)
44511312
type(c)
class 'float'
x="abc"
id(x)
35908864
type(x)
class 'Str'>
```


Scanned by CamScanner

03/01/2019

හඳුන්වන (Identifiers / names)

* Program එකක ඇති යම් යම් අංගයන් හඳුනා ගැනීම සඳහා අනුබද්ධ කළ සංකේතනාම "හඳුන්වන" ලෙස සලකයි.

* හඳුන්වන නිර්මාණය කිරීමේදී අනුගමනය කළ යුතු නීති පද්ධතියක් ඇත.
* එය පරිගණක භාෂාව මත රඳාපවතී,

පයිතන් භාෂාවේ හඳුන්වන සඳහා අදාළ නීති ...

* හඳුන්වන අක්ෂරයකින් හෝ "_" අවලක්ෂණයෙන් ආරම්භ විය යුතුය.
* පළමු අක්ෂරයට පසුව අක්ෂර හෝ 0 සිට 9 දක්වා ඉලක්කම් අඩංගු විය හැක.
* විශේෂ සංකේත අඩංගු නොවිය යුතුය. ($, @, #, *, %)
* ඕනෑම තරමක අක්ෂර සංඛ්‍යාවේ සීමාවක් නැත.
* Python මූලපද භාවිත කළ නොහැක.
* සිම්පල්, කැපිටල් භේදය ඇත.
* හඳුන්වනය තුළ හිස් අවකාශ තිබිය නොහැක.

උදා: x = 10 (x)
10 ✓

3x = 10 (x)
Syntax error

පහත සඳහන් නාමයන් අතරින් වලංගු හා වලංගු නොවන ඒවායේ නාම වෙන්කරන්න.

• වලංගු
- Name
- _name9
- _3item
- For_2
- true

• වලංගු නොවන
- 2-id_NO
- @my file
- my@ file
- True
- False

Scanned by CamScanner

දත්ත ප්‍රරූප.... (Data Types)

* කලින් නිර්වචනය කළ (එසේම එක මෙහෙයවන දත්ත අගයන් සමූහයක් "DATA TYPE" ලෙස හඳුන්වයි.
* දත්ත භාවිතයට පෙර ඒවායේ වර්ගය දැනගත යුතුය.

* පයිතන් භාෂාවේ ප්‍රධාන දත්ත වර්ග කිහිපයක් ඇත.

පයිතන් භාෂාවේ ප්‍රධාන දත්ත වර්ග

Integer
Float
long integer
String
List
Dictionary
Tuples
Set
File

පයිතන් භාෂාවේ දත්ත වර්ග 02 කි.

* Mutable - මෙම වර්ගයේ දත්ත නිර්මාණය කළ පසු ඕනෑම අවස්ථාවක වෙනස් කළ හැක.
* Immutable - වස්තූන් නිර්මාණය කළ පසු ඒවායේ අගයන් වෙනස් කළ නොහැක.

Mutable	Immutable
List	integer
Dictionaries	Float
	String

Scanned by CamScanner

වෙනස් කරන විට,
0 1 2
a=[1, 2, a]
<<< a(2)=3
a=
[1, 2, 3]

* Integer
- ධන, ඍණ සහිත නිඛිල දක්වයි.
- +/-2147483647

උදා:
A=5 A යනු නිඛිලයකි.

* Float
- දශම සහිත සංඛ්‍යා දක්වයි.

උදා:
A=45.21 A-float වර්ගයට අයත් වේ.

* long integer
- විශාල නිඛිල සංඛ්‍යා දක්වයි.
- +/-2147483647 විශාල සංඛ්‍යා

* String
- Single / Double (' '/" ") තුල දක්වයි.
- අකුරු, ඉලක්කම් හෝ වෙනත් සංකේත ඇතුලත් විය හැකිය.

උදා:
* S = "Sri lanka"
* A = 'Python'
* B = "1234"

* List (ලැයිස්තු)

- [ ] වරහන් තුල කොමාව (,) මගින් වෙන් කොට දත්ත අයිතම දක්වනු ලැබේ.
- විවිධ වර්ගයේ දත්ත අඩංගු විය හැකිය.

උදා:
[1, 2.2, 3.21, 'cow', 'horse'] ['a', 'b', 'c']

Scanned by CamScanner

* Tuple

- සරල වරහන් තුල () කොමාවෙන් (,) වෙන්කොට දක්වන ලැබේ.
- විවිධ වර්ගයේ දත්ත ඇතුලත් විය හැකිය.

උදා: ('hen', 'duck', 'rabbit', 'dog', 'car')

* Dictionary

- සඟල වරහන් තුල දක්වන ලැබේ. "{}"
- යුගල වශයෙන් ඇත.
- යුගල දැක්වීමට Colon (:) භාවිතා කරයි. යුගල වෙන්කිරීමට කොමාව (,) භාවිතා කරයි.

උදා: {'first':'alpha', 'last':'Omega'}

උදා: d={1:'a', 2:'b', 3:'c'}

d

{1:'a', 2:'b', 3:'c'}

* Set (කුලකය)

- පිළිවෙලක් රහිත දත්ත සමූහයක් ඇතුලත් කිරීමට Set භාවිත කළ හැකිය.
- එය සඟල වරහනක් තුල දක්වන ලැබේ.
- list සහ Tuple තුල ද්විත්වයක් භාවිත වුවද Set තුල ද්විත්වයක් භාවිත නොවේ.

උදා: {1,2,3,'a','b',4}

Scanned by CamScanner

10/01/2018

Variables (විචල්‍යය)

ක්‍රමලේඛ ක්‍රියා කරවීමේදී දත්තයක් තාවකාලිකව ප්‍රධාන මතකයේ තබා ගැනීම සඳහා විචල්‍යය භාවිත කරයි. පරිගණකයේ ප්‍රධාන මතකයේ නිශ්චිත කොටසකට ප්‍රවිශ වීමට උපයෝගී කරගත හැකි සංකේත නාමයක් ලෙස විචල්‍යය දැක්විය හැකිය.

Python භාෂාවේදී විචල්‍යයක් අර්ථ දක්වා, එය ධාවනය කරවීමේදී (අර්ථනිරූපණය වීමේදී) මූලික කාර්යයන් 03ක් සිදුවේ. මෙය පහත උදාහරණය මගින් පැහැදිලි කරගනිමු.

◂ a=10 යන Python ප්‍රකාශනය ධාවනය වීමේදී, පහත කාර්යයන් 3 සිදුවේ.

01. එළඹුම නිඛිලයක් ගබඩා කිරීමට හැකි ඉඩක් ප්‍රධාන මතකයෙන් (RAM) වෙන්කර ගනියි.
02. එය A ලෙස ලේබල් කරගනියි.
03. එහි අගයක 10 ලෙස යොදාගනියි.

මෙහෙයවන - Operators

දත්ත මත කළ යුතු කාර්යයන් නිරූපනය කරයි. මේවා කාණ්ඩ කිහිපයකට බෙදිය හැකිය.

- ගණිතමය මෙහෙයවන
- තාර්කික මෙහෙයවන
- සැසඳුම් මෙහෙයවන
- බිටුමත්‍ය මෙහෙයවන

• ගණිතමය මෙහෙයවන

* • + මෙහෙයවනය
• එකතුව දක්වයි.
• උදාහරණ

* 1+2=3

*10.2+6 = 16.2

*"ab"+"cd" = 'abcd'

* a=10
b=10.5
a+b
20.5

* c="abc"
d = "def"
c+d
abcdef

Scanned by CamScanner

* - මෙහෙයවනය

- අඩුකිරීම දක්වයි.
- උදාහරණ
  - *5-2=3
  - *2-5=-3
  - *10.5-3=7.5

* * මෙහෙයවනය

- ගුණකිරීම දක්වයි.
- උදාහරණ
  - *5*3=15
  - *5.0*3=15.0
  - *"abc"*5
    'abcabcabcabcabc'

*/ මෙහෙයවනය

- බෙදීම දක්වයි (/)
- උදාහරණ
  - 8/2=4.0
  - 4.0/2=2.0
  - *බෙදීමේදී සෑම විටම පිළිතුර ඉපිලෙන සංඛ්‍යාවක් ලෙස ලබාදෙයි (float) (දශමය සංඛ්‍යාවක්)

* // මෙහෙයවනය

- නිඛිල බෙදීම දක්වයි.
- උදාහරණ
  - 9//2=4
- මෙම බෙදීමේදී සෑම විටම පිළිතුර නිඛිල සංඛ්‍යාවක් ලෙස ලබාදෙයි.

* % මෙහෙයවනය

- බෙදීමට ඉතිරිවන ඉතිරිය දක්වයි.
- උදාහරණ
  - 9%2=1 / 7%4=3 / 7.0%4=3.0

Scanned by CamScanner

⊛ ** මෙහෙයවනය

* ඝාතය දක්වයි.

උදාහරණ

3 ** 2 = 9 $(3^2 = 9)$

3.** 2 = 9.0

2 ** 3 = 8

අංක ගණිතමය පරිවර්තනය

* සංඛ්‍යා 2න් එකක් සංකීර්ණ සංඛ්‍යාවක් නම්, අනෙක් සංඛ්‍යාවද සංකීර්ණ සංඛ්‍යාව ~~නම් අනෙක් සංඛ්‍යාමද~~ බවට පරිවර්තනය කරයි.
* සංඛ්‍යා 2න් එකක් ඉපිලෙන සංඛ්‍යාවක් නම්. අනෙක් සංඛ්‍යාවද ඉපිලෙන සංඛ්‍යාවක් බවට පරිවර්තනය කරයි. (float)
* සියල්ල පූර්ණ සංඛ්‍යා නම් පරිවර්තනයක් සිදුනොවේ.

තාර්කික මෙහෙයවන

* තාර්කික මෙහෙයුම්වලදී යොදාගන්නා, ද්වාර්ථයන් සාධක (Operands) බූලීය විය යුතු අතර එම ප්‍රතිඵලයද බූලීයවේ.
* සත්‍ය අගය පරීක්ෂාව

අසත්‍යය අගයන්...

None - False - '', (), [], {}

අනෙකුත් සියලු අගයන් සත්‍ය ලෙස සැලකේ.

a = {} ← false

◀ Or

True OR False = True

සාමාන්‍යයෙන් Or ද්වාරයේ ක්‍රියාකාරීත්වය අනුව සිදුවේ.

True ~~OR~~ or False = True

False ~~OR~~ or True = True

True ~~OR~~ or True = True

False ~~OR~~ or False = False

* 472 or 4710
  True
* 472
  True
* 4710
  False

~~AND~~ and

- සෑම සාමාන්‍යයෙන් AND ද්විරංගේ ක්‍රියාකාරීත්වය අනුව යයි.
- True ~~AND~~ and False = False
  False ~~AND~~ and True = False
  False ~~AND~~ and False = False
  True ~~AND~~ and True = True

(1>2) and (4>2) = False

not

- not True = False
- not False = True

සැසඳුම් මෙහෙයවන

- < වඩා අඩු
  2<3 True

[1,5] < [1,2,8] False
[1,5] < [1,6,8] True
"a">"b" False

- > වඩා වැඩි
  2>3 False
  'c'>'d' False
  [1,5]>[1,2,8] True

බිටුමත් මෙහෙයවන

* බිටුමය මෙහෙයවන ද්විමය ආකාරයෙන් දත්ත භාවිතා කරයි.

* බිටුමය OR (|)

- 10|12
  14
- 5|4
  5

10 → 1010
12 → 1100
1110 → 14

5 → 101
4 → 100
101 → 5

බිටුමය AND (&)

* 10&12
8

```
 1010 → 10
 1100 → 12
 ----
 1000 → 8
```

* 5& 4
4

```
5 → 101
4 → 100
    ---
    100 → 4
```

බිටුමය XOR (^)

* 10^12

```
10 → 1010
12 → 1100
     ----
     0110 → 6
```

Left Shift (<<)

* a=10
a<<2
40

```
    | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
```

32, 8, $2^1 \times 0$

32+8 - 40//

~~a=10~~
* a<<1
20

Right Shift (>>)

a>>1
5//

```
| 1 | 0 | 1 | 0 |
```

10

17/01/2018

බිටුමය NOT

බිටුමය NOT මගින් දෙකෙහි අනුපූරක ආකාරයෙන් පිළිතුර ලබාදේ.

~10 → -11

~10
~1010 ← 10
0101

11 → 1011
-11 → 0100
1+
-11 → 0101
(2හි අනුපූරකයෙන්)

* ~12 → -13
* ~13 → -14

~12
1100
0011

13 → 1101
-13 → 0010
1+
0011

* ~20 = -21

2	20
2	10 - 0
2	5 - 0
2	2 - 1
2	1 - 0
	0 - 1

20 → 10100
01011
~~10100~~

~~21 → 10101~~
~~01010~~
~~1~~
~~01011~~

21 → 10101
01010
1
01011

2	21
2	10 - 1
2	5 - 0
2	2 - 1
2	1 - 0
	0 - 1

තන්තු (String)

- තනි පේළියක String දැක්වීමේදී Single Quotes (') හෝ Double Quotes (") භාවිතා කරනු ලබයි
- පේළි එකකට වඩා වැඩි වාක්‍ය සඳහා Triple Quotes (''') භාවිතා කරයි. (Single Quotes 3/ Double Quotes 3)

```
a = '''abc
def
ghi'''
a
'abc\ndef\nghi'
```

Scanned by CamScanner

\n - New line Character

* නව පේළියකට ගමන් කිරීම සඳහා භාවිත කරයි.

උදාහරණ :

```
Print ("Sri\nLanka")
Sri
Lanka
```

Simple →

```
Print("A\nB\nC\nD")
A
B
C
D
```

Single Quotes and Double Quotes

Single/Double Quotes මුද්‍රණය කර ගැනීමට Black Slash (\) යොදා ගත හැක.

```
Print ("\'Sri Lanka\' is Very beautiful Country")
'Sri Lanka' is Very beautiful Country
```

```
Print ("\"Sri Lanka\" is Very beautiful country")
"Sri Lanka" is Very beautiful Country.
```

```
* Print(" 'abc' ")
'abc'
```

```
* Print(' "abc" ')
"abc"
```

* පිටතින්ම Single Quotes යොදනවා නම් එවිට Double Quotes තුළද, Double Quotes යොදනවා නම් එවිට Single Quotes තුළද දැක්විය හැකිය.

Scanned by CamScanner

\t - Horizontal Tab

මෙමගින් tab එකක් වැටේ.

```
Print ("Sri\tLanka")
Sri	Lanka
```

```
* Print ("Sri\t Lanka \n Sri \t Lanka \n Sri \t Lanka")
```

Output එක වෙනස

```
Sri	Lanka
Sri	Lanka
Sri	Lanka
```

\\ - for insert "\"

```
Print ("Sri Lanka\\Ceylon ")
Sri Lanka\Ceylon
```

```
Print ("A\\\B")
A\\B
```

```
Print ("A\\B")
A\B
```

වාක්‍ය 1 හෝ 2 ක් එකතු කිරීම. (String Concatenation)

```
Print ("Sri" "Lanka")
SriLanka
```

```
Print ("Sri" "	Lanka")
Sri	Lanka
```

```
S = "Sri"
Ws = "  "
l = "Lanka"
C = S + Ws + l
Print(C)
Sri  lanka
```

Scanned by CamScanner

Math With String (ගණිතය සමග තන්තු)

* තන්තුවක් නිඛිලයකින් ගුණ කල හැකිය.

```
Print("Srilanka"*3)
Srilanka Srilanka Srilanka
```

Length of a String (තන්තුවක දිග)

* වාක්‍යයක ඇති අකුරුවල ගණන දක්වයි.
* හිස්තැන (Space) ද අකුරක් ලෙස සලකයි.

```
>>> S = "Sri Lanka is Very beautiful Country"
>>> len(S)
35
```

(Space එකත් අකුරකි)

```
S = "abcdefghijkl"
len(S)
12
```

String Index (තන්තුවක සූචිය)

- තන්තුවක සූචිය ආකාර 2ක් වෙතිය.
- තන්තුවේ වමේ සිට දකුණු පසට 0 න් ආරම්භ වී වැඩිවන සූචියක් වෙතිය.

```
      -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
S =    a   b   c   d  e  f  g  h  i  j  k  l
       0   1   2   3  4  5  6  7  8  9  10 11
```

```
S=[0]
'a'
```

```
S=[-2]
'k'
```

Scanned by CamScanner

* එකතුවෙහි දකුණු පසින් ආරම්භ වී -1න් ආරම්භ වී - අගය අඩුවන ලෙස සුචියන් පවතියි.

උදා: S = "ABCDEFGHIJ"

(indices 0 1 2 3 4 5 6 7 8 9; I = -2, J = -1)

* S[0]
'A'

* Print(S[0])
A

* Print(S[3])
B

* Print(S[-1])
J

* Print[S[-4]]
Type error

* Print(S[-4])
G

* Print(S[2:5])
CDE

* S[5:25]
'FGHIJ (තිබෙන ප්‍රමාණය පෙන්වයි)

* Print(S[3:8])
DEFGH

* S[25]
IndexError

* S[3:]
'DEFGHIJ'

* S[:3]
'ABC'

* S[6:2]
''

* S[2::]
'CDEFGHIJ'
↑ එතැන් සිට ගමන් කරයි.

* S[2::2]
'CEGI' ↖ 2න් 2ට ගමන් කරයි.

Scanned by CamScanner

* S[1: :3]
'BEH'

* S[: :3] ← මුල ඉඳන් 3න් 3ට ගමන් කරයි.
'ADGJ'

* S[1:6: :2]
SyntaxError: invalid syntax

LIST - ලැයිස්තු

* කොටු වරහන් තුල කොමා, එකින් එක වෙන්කිරීමෙන් ලැයිස්තුවකට අයිතම ලබා දිය හැකිය.

* ලැයිස්තුවක් තුල තවත් ලැයිස්තු ඇතුලත්විය හැකිය
* ලැයිස්තුවක් තුල විවිධ වර්ගයේ දත්ත වර්ග (data types) ඇතුලත් විය හැකිය.

```
L=["Saman","Kamal", 25, 123.4]
```

Position Of List

```
L=["Saman","Kamal", 25, 123.4]
L[0]
'Saman'
Print(L[1])
Kamal
L[2]
25
```

• ලැයිස්තු තුලද තන්තුවල ලෙසම ස්ථානයක් පවතියි.

24/01/2019

තන්තුවක ලෙසම ලැයිස්තුවකද ස්ථානයක් ඇත. එය වම්පස සිට දකුණු එකට 0න් ආරම්භව 1,2,3... ලෙස වැඩිවන ආකාරයෙන්ද, දකුණුපස සිට වම්පසට

Scanned by CamScanner

-1,-2,-3.... ලෙස වැඩිවන පිළිවෙලටද ඇවතිය.

```
      -9  -8   -7   -6    -5      -4      -3   -2   -1
L = ['a','b', 'c', 'd', 'efg', [1,2,3], 'h', 'i', 'j']
      0   1    2    3     4       5       6    7    8
```

- L[0] = 'a'
- L[5] = [1,2,3]
- L[4] = 'efg'
- L[3:7] = ['d', 'efg', [1,2,3], 'h']
- S = L[3:7]
- Print(S)
  ['d', 'efg', [1,2,3], 'h']
- L[4:6] = ['efg', [1,2,3]]
- L[:4] = ['a','b','c','d']

- Print (L[1] * 3)
  bbb
- Print(L[5] * 2)
  [1,2,3, 1,2,3]

- L[4:] = ['efg', [1,2,3], 'h', 'i', 'j']
- L[2::2] = ['c', 'efg', 'h', 'j']
- L[2:8::2] = Syntax error
- L[::3] = ['a', 'd', 'h']
- L[4][1] = 'f'
  ↳ [efg]
  0 1 2
- L[5][2] = 3
- L1 = L[:4]
  L2 = L[4:7]
  Print(L2 + L1)
  ~~['a','b','c','d'] + ['efg',[1,2,3],'h']~~
  ['efg', [1,2,3], 'h', 'a', 'b', 'c', 'd']

ලැයිස්තුවක දිග

len(L)
9 // (ලැයිස්තුවක ඇති අයිතම ගණන)

Tuples

- සරල වරහනක් () කොමාවෙන් (,) වෙන්කොට දක්වනු ලබයි.
- එකම වර්ගයේ දත්ත ඇතුලත් විය හැකිය.

උදා: ('Then', 'duck', 'rabbit', 'cow')

- Tuple එකක් ලැයිස්තුවකට බොහෝ දුරට සමානවේ, එහෙත් ප්‍රධාන වෙනස්කම් 02ක් දැකිය හැකිය.
  - * ලැයිස්තුවෙහි කොටු වරහන් භාවිත වුවද Tuple එකෙහි සුළුවරහන් භාවිතාවේ.
  - * ලැයිස්තුව mutable වන අතර Tuple immutable වේ.

Scanned by CamScanner

* වචන හා ලැයිස්තුවල මෙන්ම tuple එකේද සූචියක් ඇත.

t=(1,2,3,4,5)
t[0]=1
t[3]=4

SET

* Set එකක් කොමා වලින් වෙන්කර ලියනු ලබයි.
* මෙහි පිළිවෙලක් නොමැත.
* සූචියක් නැත.

```
S={12,2,2,4,5,4}
S
{12,2,4,5}
```

◀ පරිශීලක දත්ත ලබාගැනීම.
~~මේසඳහා Input ශ්‍රිතය භාවිතා කරයි.~~

• වචනමය ආකාරයෙන් ලබාගැනීම සඳහා,
* මේසඳහා Input ශ්‍රිතය භාවිතා කරයි.

```
X=input("Enter your name")
```

සැමවිට පිළිතුර වචන (String) ආකාරයෙන් ලැබේ.

• පූර්ණ ආකාරයෙන් ලබාගැනීම සඳහා,

```
X=int(input("Enter your Age"))
```

```
A=input("Enter your name")
~~B=int(input("Enter~~
Enter your name Asith
B=int(input("Enter your age")
Enter age 20
Print("Your name",a)
Your name Asith
```

```
A=input("Enter you name")
B=int(input("Enter you age"))
Print("Your name is",a)
Print("Your Age is",B)
```

Scanned by CamScanner

- සෘජුකෝණාස්‍රයක දිග (length) හා පළල (width) ඇතුළත් කොට එහි වර්ගඵලය (Area) ගණනය කිරීමට ක්‍රමලේඛයක් ලියන්න.

```
A= ~~input~~ int(input("length"))
B= int(input("width"))

Print("Area"; A*B)
```

```
L= int(input("length"))
W= int(input("width"))
area = L* W
Print("Area:", area)
```

- භාණ්ඩයක මිල ලබාගෙන එයට 10% ක වට්ටමක් ලබාදුන් පසු නව මිල ලබාගැනීම සඳහා Python ක්‍රමලේඛයක් ලියන්න.

~~A = int(input("Price"))~~

~~New Price =~~ [illegible]

```
Price=int(input("price"))
Price = Price * 90/100
print ("NewPrice with discount :", Price)
```

Scanned by CamScanner