

A/L ICT



2019 onwards

New syllabus

Advanced Level

Sri Lankan

Arithmetic Operators

எண்கணித செய்பணிகள்

- + :- Addition கூட்டல்
- - :- Subtraction கழித்தல்
- * :- Multiplication பெருக்கல்
- / :- Division வகுத்தல்
- //:- Integer division வகுத்தல்(முழு எண்)
- ** :- Exponentiation வலு
- % :- Modulus மீதி

❖ Example

```

ore.py - C:/Users/L.Visagan/Desktop
File Edit Format Run Options
print(5+2)
print(5-2)
print(5*2)
print(5/2)
print(5**2)
print(5%2)

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:/Users/L.Visagan/Desktop/ore.py -----
7
3
10
2.5
25
1
>>>
Ln: 12 Col: 4
Ln: 8 Col: 0

```

Relational / Comparison operators

தொடர்புநிலை / ஒப்பீட்டு செய்பணிகள்

- $>$:- Greater than
- $<$:- Less than
- $> =$:- Greater than or equal to
- $< =$:- Less than or equal to
- $= =$:- Equal
- $! =$:- Not equal

❖ Example

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(5>2)
True
>>> print(5>=2)
True
>>> print(5<2)
False
>>> print(5<=2)
False
>>> print(5==2)
False
>>> print (5!=2)
True
>>> |
```

❖ Example

Print $(5 < = 5)$ \longrightarrow True

Logical Operators

தருக்க செய்பணிகள்

- 1) AND
- 2) OR
- 3) NOT

❖ Example

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (I
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(5>3 and 6<4)
False
>>> print (5>3 or 5<2)
True
>>> print ( not ( 5>4))
False
>>> |
```

Shift Operators

- >> :- right-shift
- << :- left- shift

❖ Example

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(5>>2)
1
>>> print(5<<2)
20
>>> |
```

• $5 \gg 2 \longrightarrow 101 | 2 \xrightarrow{\leftarrow} 1_2 \longrightarrow 1_{10}$

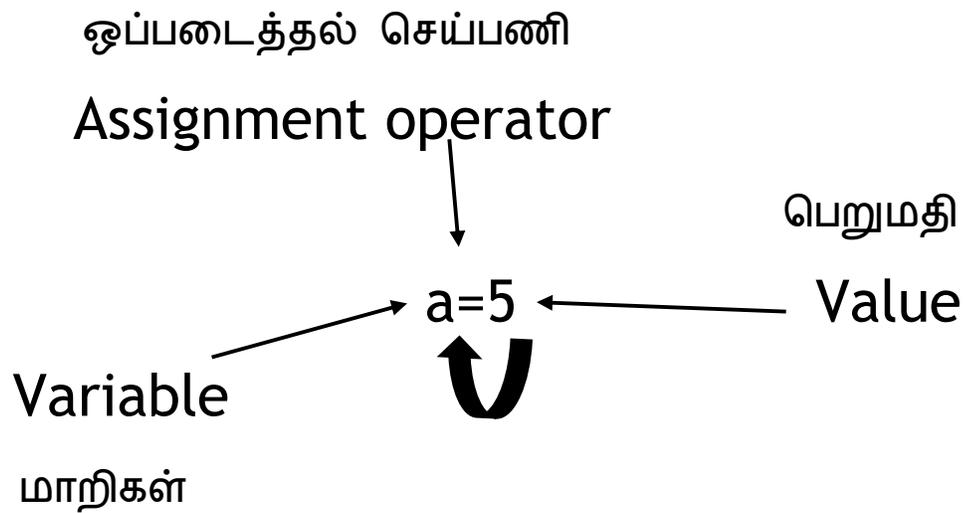
• $5 \ll 2 \longrightarrow 101 | 2 \xrightarrow{\longrightarrow} 10100_2 \longrightarrow 20_{10}$

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(12<<3)
96
>>> print(12>>3)
1
>>> |
```

Variables

மாறிகள்

- In programming, variables are used to store data.
- கணினி செய்நிரலாக்கத்தில் தரவுகளை சேமிப்பதற்கு மாறிகள் பயன்படுத்தப்படுகின்றன.



- Memory space is allocated for storing an integer value.
- The value " 5 " is assigned to variable " a " and stored in the memory location.

- விளக்கம்:- முழு எண் பெறுமதியினை சேமிக்க நினைவக வெளி ஒன்று ஒதுக்கப்பட்டுள்ளது.
- 5 எனும் பெறுமதி மாறியிற்கு ஒப்படைக்கப்பட்டுள்ளது.
- 5 எனும் பெறுமதி மாறியிற்கு ஒப்படைக்கப்பட்டு அந்நினைவக இட அமைவில் சேமிக்கப்படுகிறது.

❖ Example

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=2
>>> a=a+2
>>> print(a)
4
>>> a=2
>>> b=3
>>> c=a+b
>>> print(c)
5
>>> |
```

❖ Explanation

- A memory location for variable "a" is created.
- The value 2 is assigned to variable "a" and stored in memory.
- A memory location for variable "b" is created.
- The value 3 is assigned to variable "b" and stored in memory.
- The values stored in variable "a" and "b" are retrieved from memory and added together and assigned to variable "c" and stored in memory.
- மாறி a இற்கான நினைவகத்திற்கு ஒர் இடம் உருவாக்கப்படுகிறது.
- பெறுமதி 2 மாறி a ற்கு ஒப்படைக்கப்பட்டு அந்நினைவக இடஅமைவில் சேமிக்கப்படுகிறது.
- மாறி b இற்கான நினைவகத்திற்கு ஒர் இடம் உருவாக்கப்படுகிறது.

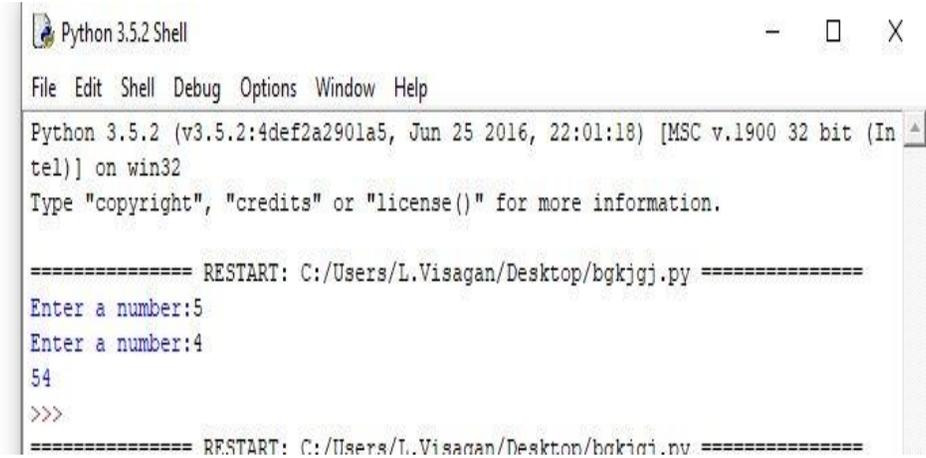
- பெறுமதி 3 மாறி b ற்கு ஒப்படைக்கப்பட்டு அந்நினைவக இட அமைவில் சேமிக்கப்படுகிறது.
- மாறிகள் a,b ஆகியவற்றில் சேமிக்கப்பட்ட பெறுமதிகள் நினைவகத்திலிருந்து மீள பெறப்பட்டு அவற்றின் பெறுமதிகள் ஒன்றாக கூட்டப்பட்டு மாறி c ற்கு ஒப்படைக்கப்பட்டு அதன் நினைவக இடஅமைவில் சேமிக்கப்படுகிறது.

input () function

- input () function is used to enter user input through keyboard.
- It takes value as a string data type.
- input () செயலி விசைப்பலகை மூலம் பயனர் உள்ளீட்டினை உள்ளீடு செய்ய பயன்படுகிறது.
- உள்ளீடு செய்யப்படுகின்ற பெறுமதி ஒரு வரியுருச்சக்கரம் தரவு வகையாக பெறப்படுகின்றது.

❖ Example

```
a=input("Enter a number:")  
b=input("Enter a number:")  
print(a+b)
```



```
Python 3.5.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
  
===== RESTART: C:/Users/L.Visagan/Desktop/bgkjgj.py =====  
Enter a number:5  
Enter a number:4  
54  
>>>  
===== RESTART: C:/Users/L.Visagan/Desktop/hakiaj.py =====
```

Data type conversion

தரவு வகை மாற்றீடு

❖ String → Integer

```
a=int(input("Enter a number:"))
b=int(input("Enter a number: "))
print(a+b)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/jjjjj.py
Enter a number:4
Enter a number:5
9
>>> |
```

❖ String → Float

```
a=float(input("Enter a number:"))
b=float(input("Enter a number: "))
print(a+b)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/jjjjj.py
Enter a number:5.0
Enter a number:4.1
9.1
>>>
```

❖ Concatenation

❖ சேர்த்தல்

```
x=3  
y=2  
print("x+y=", (x+y))
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/jjjjj.py  
x+y= 5  
>>> |
```

❖ Explanation

X=3

Y=2

Print ("x+y=", (x+y))

String

Numeric value

Concatenation operator

```
length=int(input("Enter a length of a square:"))
print("perimeter=" , 4*length)
print("Area=" , length * length)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/jjxxx.py
Enter a length of a square:10
perimeter= 40
Area= 100
>>> |
```

```
length=int(input("Enter length :"))
breadth=int(input("Enter breadth :"))
print("perimeter=" , 2*(length+breadth))
print("Area=" , length * breadth)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/jjxxx.py
Enter length :10
Enter breadth :20
perimeter= 60
Area= 200
>>> |
```

Conditional statements

நிபந்தனை கூற்றுகள்

if condition:

Statement 1

else:

Statement 2

- Program checks the condition.
- If the condition is true, then the statement 1 is executed.
- Otherwise statement 2 is executed.

- செய்நிரல் முதலில் நிபந்தனையை சரிபார்க்கின்றது நிபந்தனை உண்மை எனின் கூற்று 1 நிறைவேற்றப்படுகிறது
- இல்லாவிடின் கூற்று 2 நிறைவேற்றப்படும்.

```
marks=int(input("Enter marks :"))
if marks >90:
    print( "Excellent")
else:
    print( "Good")
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter marks :85
Good
>>>
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter marks :96
Excellent
>>> |
```

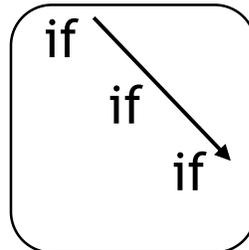
```
num=int(input("Enter a number :"))
if num%2 == 1:
    print( "Odd number")
else:
    print( "Even number")
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number :5
Odd number
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number :16
Even number
>>> |
```

```
n1=int(input("Enter a number :"))
n2=int(input("Enter a number :"))
if n1>n2:
    print( n1,"is big")
else:
    print( n2 , "is big")
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number :10
Enter a number :25
25 is big
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number :55
Enter a number :23
55 is big
>>> |
```

Nested if



```
a=int(input("Enter a number:"))
b=int(input("Enter a number:"))
c=int(input("Enter a number:"))
if a>b:
    if a>c:
        print(a)
    else:
        print(c)
else:
    if b>c:
        print(b)
    else:
        print(c)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number:55
Enter a number:45
Enter a number:35
55
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number:54
Enter a number:125
Enter a number:23
125
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/xyz.py =====
Enter a number:56
Enter a number:96
Enter a number:150
150
>>> |
```

```
marks=int(input("Enter a subject marks:"))
if marks > 100 or marks < 0:
    print("Invalid marks, Try again")
elif marks >= 75:
    print("A")
elif marks >= 50:
    print("C")
elif marks >= 35:
    print("Pass")
else:
    print("W")
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter a subject marks:55
C
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter a subject marks:101
Invalid marks, Try again
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter a subject marks:-56
Invalid marks, Try again
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter a subject marks:25
W
>>> |
```

```

a=int(input("Enter subject marks 1 :"))
b=int(input("Enter subject marks 2 :"))
c=int(input("Enter subject marks 3 :"))
total=a+b+c
average=total/3
print("Total marks :",total)
print("Average marks :",average)
if average >= 90:
    print ("Excellent")
else:
    print ("Try again")

```

```

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter subject marks 1 :85
Enter subject marks 2 :78
Enter subject marks 3 :69
Total marks : 232
Average marks : 77.33333333333333
Try again
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter subject marks 1 :85
Enter subject marks 2 :82
Enter subject marks 3 :90
Total marks : 257
Average marks : 85.66666666666667
Try again
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter subject marks 1 :95
Enter subject marks 2 :96
Enter subject marks 3 :97
Total marks : 288
Average marks : 96.0
Excellent
>>>

```

❖ Electricity Meter Reading Question

- 50 units:- 300/=
- 51 - 100 units:- 5/= per each
- >100 units :- 20/= per each

```

pm=int(input(" Enter previous meter reading :"))
cm=int(input(" Enter current meter reading :"))
u=cm-pm
print(" No of units :",u)
if u<0:
    print("Please check meter reading")
elif u<=50:
    print("Charge :- ",300)
elif u <= 100:
    print("Charge :- ",300+(100-u)*5)
elif u > 100:
    print ("Charge :- ", (300+50*5+(u-100)*20))

```

```

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter previous meter reading :125
Enter current meter reading :100
No of units : -25
Please check meter reading
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/eee.py =====
Enter previous meter reading :256
Enter current meter reading :386
No of units : 130
Charge :- 5150
>>>

```

Operations precedence

செய்ப்பணிகளின் முன்னுரிமை

- $\text{print}(5*3+2)$

1
2
- $\text{print}(5*2-4/2+2*2)$

1
4
5
3
2
- $\text{print}(7\%2+5)$

1
2
- $\text{print}(3**2+1)$

1
2
- $\text{print}(3**2+4/2-1)$

1
4
2
3

Bitwise AND, OR, XOR

❖ Bitwise AND

- & :- Ampersand

print (5 & 4) = 4 (output)

101 & 100	101	
	100	
	100	
	101	→ 4

❖ Bitwise OR

- | :- Vertical bar

print(5 | 4) = 5 (output)

101 100	101	
	100	
	101	→ 5

❖ Bitwise XOR

- ^ :- Caret (Hat)

print (5 ^ 4) = 1 (output)

101 ^ 100	101	
	100	
	001	→ 1

Combined Assignment Symbols

- `+=`
- `-=`
- `*=`
- `/=`
- `//=`
- `**=`
- `%=`

❖ Examples

```
a=5
a+=2 → a= a+2
print(a)
Output: 7
```

```
a=8
a%=3 → a= a % 3
print(a)
Output: 2
```

```
a=2
a**=3 → a= a**3
print(a)
Output: 8
```

```
a="python"
```

```
a+="is not a snake"
```

```
print(a)
```

Output: python is not a snake

Identifiers

அடையாளப்படுத்திகள் / அடையாளங்காணிகள்

Valid identifiers வலிதான அடையாளப்படுத்திகள்	Invalid identifiers வலிதற்ற அடையாளப்படுத்திகள்
a ab a_b _a a_ a2	a b → No space a-b a#b a@b a\$b 2a-Does not start with digit if for else while No special characters No keywords

Key words OR Reserved Words

சாவிச்சொற்கள் / ஒதுக்கச்சொற்கள்

❖ Keywords cannot be used as ordinary Identifiers.

and	elif	pass	exec
del	global	yield	in
from	rr	break	ralse
not	with	except	continue
while	assert	import	finally
as	else	print	is
try	if	class	return
lambda		for	def

❖ Integer முழு எண்

```
a=3
print (a)
```

❖ Float தசம எண்

```
a=3.4
print (a)
```

❖ Boolean (capital must)

```
a=True
print(a)
```

```
a=False
print(a)
```

❖ String

```
a="python"  
print(a)
```

OR

```
a='python'  
print(a)
```

Output: python

❖ Note

```
a='nimal's address'  
print(a)
```

ERROR

```
a="nimal's address"  
print (a)
```

Output: nimal's address

List

- a= [3, 5, 6, 1, 9]
print(a)

Output: [3, 5, 6, 1, 9]

- a= [3, 5, 6, False, 'Python']
print(a)

Output: [3, 5, 6, False, 'Python']

- a=[2, 4, [1,5], 9]
print (a)

Output: [2, 4, [1, 5], 9]

- `a= [[]]]`
`print (a)`

Output: [[]]]

- `a=[2,3]`
`b= [1, 8, 7]`
`print (a+b)`

Output: [2, 3, 1, 8, 7]

- `a=[4,5,7,2,3,9]`
`print (a [3])`

Output: 2

- `a= (4, 5, 7, 2, 3, 9)`
`print (a [-2])`

Output: 3

- `a=[4,5,7,2,3,9]`
`print (a [1:5])`

Output: [5, 7, 2, 3]

- `a=[4,5,7,2,3,9]`
`print (a [2 :])`

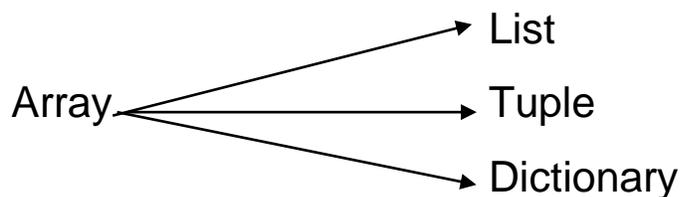
Output: [7, 2, 3, 9]

- `a=(4,5,7,2,3,9)`
`print (a (:3))`

Output: [4, 5, 7]

- `a= [4, 5, 7, 2, 3, 9]`
`print (a [:3])`
Output: `[4, 5, 7]`
- `a=[4,5,7,2,3,9,1,8]`
`print (a [1::2])`
Output: `[5, 2, 9, 8]`
- `a=[4,5,7,2,3,9,1,8]`
`print (a [::-2])`
Output: `[4, 7, 3, 1]`
- `a=[4,5,7,2,3,9,1,8]`
`print (a [1:6:2])`
Output: `[5, 2, 9]`
- `a= [4,5,7,[2,3,9],1,8)`
`print (a [3][2])`
Output: `9`
- `a=[4,5,7]`
`a [1] = 6`
`print (a)`
Output: `[4, 6, 7]`

❖ Note



Dictionary

- Dictionary must be in pair.
- `d={key: value}`
- `d={4:'Apple',1:'Orange',3:'Banana'}`
Print (d)
Output: {1: 'Orange', 3: 'Banana', 4: 'Apple'}
- `d={4:'Apple',1:'Orange',3:'Banana'}`
`print(d [3])`
Output: Banana
- `d={4:'Apple',1:'Orange',3:'Banana'}`
`d [1] = 'Strawberry'`
`print (d)`
Output: {1:'Strawberry', 3:'Banana', 4:'Apple'}
- `d = {'A':'Apple','b':'Ball'}`
`print (d ['A'])`
Output: - Apple

Computer Programming Languages

கணினி செய்நிரலாக்க மொழிகள்

- C
- C++
- Python
- Java
- C#
- Pascal
- FORTRAN
- COBOL

Generations of computer programming Languages

- 1GL
- 2GL
- 3GL
- 4GL

❖ 1GL

- Also called machine Language.
- Programs are written in binary/machine code.(1,0)
- Execution of programs is very fast.
- No programs translation needed.
- Difficult to write/test programs comparing with 3GL/4GL.
- Tied up with the computer architecture.
- இது இயந்திரமொழி என அழைக்கப்படுகிறது

- செய்நிரல்கள் யாவும் இயந்திரக்குறிமுறையில் (0,1) எழுதப்படுகின்றன.
- செய்நிரல்களின் நிறைவேற்றுகை மிக வேகமானது.
- செய்நிரல்களின் நிறைவேற்றுகைக்கு மொழிமாற்றம் அவசியமில்லை.
- 3GL, 4GL ஆகியவற்றுடன் ஒப்பிடும் போது செய்நிரலிகளை எழுதுதல் சோதனை செய்தல் கடினமானது.
- கணினி கட்டமைப்பில் நேரடியாக தொடர்புபட்டு தங்கியுள்ளது.

Binary

10001101
10101110

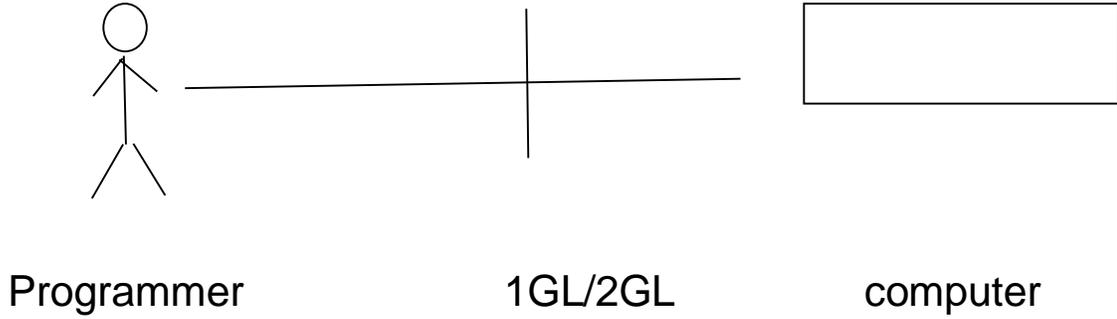
❖ 2GL

- Also called Assembly Language.
- Programs are written in mnemonics instructions
- Used to write programs for CPU.
- Assembler is needed to execute programs
- Difficult to write/test programs.
- Tied up with the computer architecture.
- Mapping between assemble instructions and machine instructions is 1-1.

- ஒருங்குசேர்ப்பி மொழி எனப்படும்
- செய்நிரல்கள் யாவும் குறியீடுகளின் அடிப்படையில் அமைந்த அறிவுறுத்தல்களில் எழுதப்படுகிறது.
- CPU விற்கான செய்நிரலாக்கத்தின் போது இது பயன்படுத்தப்படுகிறது.
- இதில் எழுதப்படுகின்றது செய்நிரல்களின் நிறைவேற்றுகைக்கு ஒருங்குசேர்ப்பி அவசியமாகும்
- இது கணினி கட்டமைப்புடன் நேரடியாக தொடர்புபட்டது
- ஒருங்கு சேர்ப்பி மொழியால் எழுதப்பட்ட அறிவுறுத்தல்கள் மற்றும் இயந்திர மொழி அறிவுறுத்தல்கள் ஆகியவற்றுக்கு இடையிலான படமிடல் 1-1 ஆகும்.
- செய்நிரல்களை எழுதுதல் சோதனை செய்தல் கடினமானது.

ADD 2, 3, R ₁
SUB 2, 4, R ₂

❖ Note



- 1GL and 2GL are collectively called 'Low Level programming Languages'.
- 1GL, 2GL என்பன (Low level programming languages) தாழ்மட்ட கணினி செய்நிரலாக்க மொழிகள் எனப்படும்.

❖ 3GL

- Example: C, C++, Java, Pascal.
- Programs are written using mathematical symbols and natural Language words.
- Execution of programs is slow.
- Valid program translator needed to execute programs.
- Easy to write/test programs in comparing with 1GL/2GL.
- செய்நிரல்கள் யாவும் கணித குறியீடுகள் மற்றும் இயற்கை மொழியிலமைந்த சொற்கள் போன்றவற்றை பயன்படுத்தி எழுதப்படுகின்றது.
- செய்நிரல்களின் நிறைவேற்றுக்கு கதி குறைவானது.

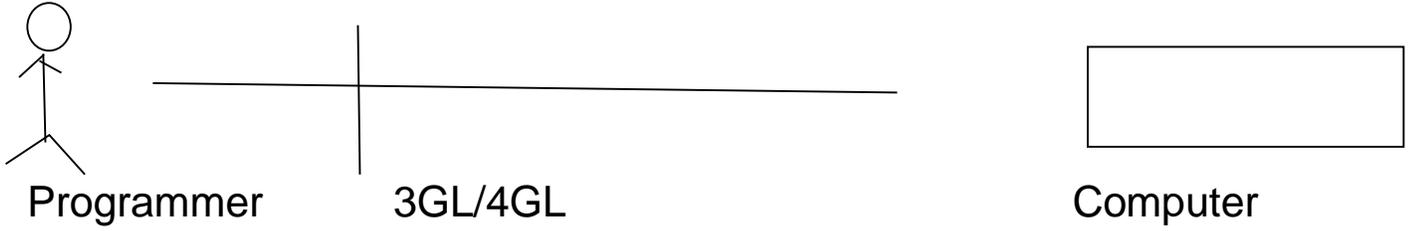
- செய்நிரல்களின் நிறைவேற்றுக்கு வலிதான செய்நிரல் மொழிமாற்றி அவசியமாகும்.
- 1GL, 2GL ஆகியவற்றுடன் ஒப்பிடும் போது செய்நிரலை எழுதுதல் சோதனை செய்தல் இலகுவானது.

❖4GL

- Example – SQL
- Programs are written using natural Language word.
- Execution of programs is very slow.
- Valid program translator needed.
- Easy to write/test programs comparing with 1GL/2GL
- 4GL is a Language and its environment that is used to develop software rapidly.
- செய்நிரல்கள் யாவும் இயற்கை மொழியிலமைந்த சொற்களை பயன்படுத்தி எழுதப்படுகிறது.
- செய்நிரல்களின் நிறைவேற்றுகை கதி மிகவும் குறைந்தது.
- செய்நிரல்களின் நிறைவேற்றுகைக்கு வலிதான செய்நிரல் மொழிமாற்றிகள் அவசியம்.
- 1GL, 2GL போன்றவற்றுடன் ஒப்பிடும் போது செய்நிரல்களை எழுதுதல் சோதனை செய்தல் இலகுவானது.

- 4GL என்பது கணினி செய்நிரலாக்க மொழி மற்றும் அதன் சூழலினை குறிக்கின்றது.
- மென்பொருட்களை மிக வேகமாக அபிவிருத்தி செய்ய பயன்படுத்தப்படுகிறது.

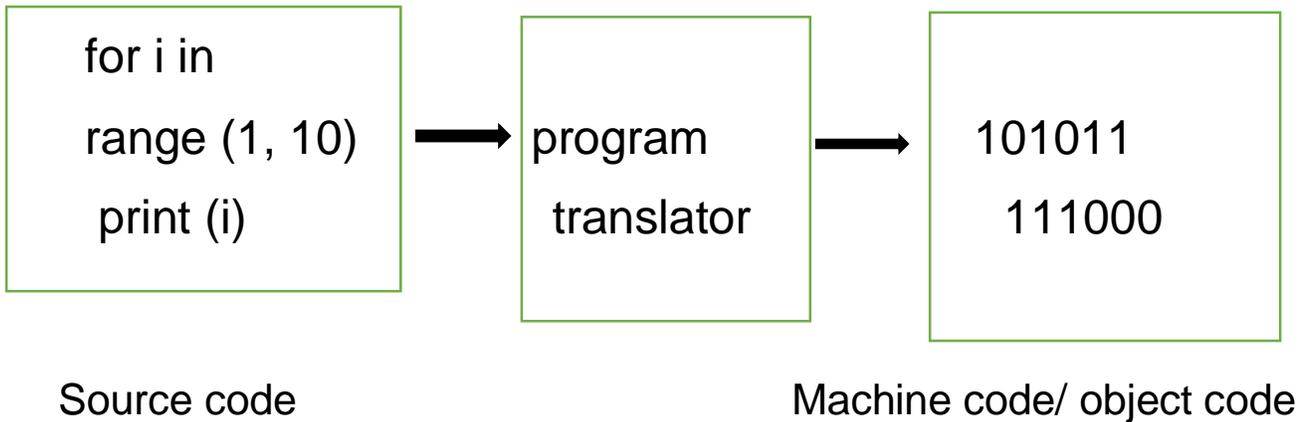
❖ Note



- 3GL and 4GL collectively called 'High Level programming Languages'.
- 3GL and 4GL உயர் கணினி செய்நிரலாக்க மொழிகள் என்று அழைக்கப்படுகின்றன.

Program Translators

செய்நிரல் மொழிமாற்றிகள்



- A program translator is a computer program that performs the translation of a program written in a given programming language into machine understandable code/object code.
- செய்நிரல் மொழி மாற்றி என்பது ஓர் கணினி செய்நிரலாகும். அது தரப்பட்ட கணினி செய்நிரலாக்க மொழியில் எழுதப்பட்ட செய்நிரல்களை இயந்திர குறிமுறைக்கு , இலக்கு குறிமுறைக்கு மாற்றுகின்றது.

❖ Three types of Translators

- **Compiler**
- **Interpreter**
- **Assembler**

❖ Compiler

❖ தொகுப்பி

- Eg:-Java, c, c++

for i in

range (1, 10) → compiler → 101011
print (i) 111000

- A compiler translates source code into object code.
- First the entire program is converted into machine code at a time.
- Then, the compiled code can be executed on the computer.
- தொகுப்பியானது மூலக்குறிமுறையினை இலக்கு குறிமுறையாக மாற்றுகிறது.
- முதலில் முழுசெய்நிரலும் ஒரே நேரத்தில் இயந்திரக் குறிமுறைக்கு மாற்றப்படுகிறது பின்னர் தொகுக்கப்பட்ட குறிமுறை கணினியில் நிறைவேற்றப்படுகிறது.

❖ Interpreter

❖ வரிமொழிமாற்றி / பொருள்கோடலி

- Eg:- Python

for i in

range (1, 10) → Interpreter → 101011

print (i) 111000

- An interpreter translates and executes source code.
- One instruction at a time, is converted into machine code and executed by the processor.
- வரிமொழிமாற்றி மூலக்குறிமுறையான இயந்திரக் குறிமுறைக்கு மொழிமாற்றி நிறைவேற்றுகின்றது.
- ஒரு நேரத்தில் ஒரு அறிவுறுத்தல் இயந்திரக்குறிமுறைக்கு மாற்றப்பட்டு கணினியில் நிறைவேற்றப்படுகிறது.

❖ Assembler

❖ ஒருங்குசேர்ப்பி

- A computer program which translates assembly language mnemonics to an object code/machine code format.
- Assembler uses either compiler or interpreter fashion.
- Assembler is used in 2GL.
- ஒருங்குசேர்ப்பி என்பது ஒருங்குசேர்ப்பி மொழியில் எழுதப்பட்ட செய்நிரல் குறியீடுகளை இயந்திரக் குறிமுறைக்கு மாற்றுகின்றது.
- ஒருங்குசேர்ப்பி ஆனது தொகுப்பி வரிமொழிமாற்றி(பொருள்கோடலி) ஆகிய ஏதேனும் ஒன்றின் பண்பை கொண்டதாகும்.
- 2ம் தலைமுறைக்(2GL) கணினி மொழியில் ஒருங்குசேர்ப்பி பயன்படுத்தப்படுகிறது.

Looping structure

தடக்கட்டமைப்பு

- If a program or a part of program is repetively executed it is called “loop”.
- Eg: - while loop, for loop.
- செய்நிரல் ஒன்று / செய்நிரலின் பகுதி ஒன்று மீளமீள நிறைவேற்றப்படுமெனின் அது தடம் எனப்படும்.

While loop

- While (condition)
Statements
- Program first check the condition.
- If the condition is true, statements are executed.
- Otherwise it terminates the loop.

- முதலில் நிபந்தனையினை சரிபார்க்கும்.
- நிபந்தனை உண்மை எனில் கூற்றுக்கள் நிறைவேற்றப்படும்.
- நிபந்தனை பொய் எனின் தடத்தினை விட்டு வெளியேறும்.

```
i=1
while (i<5):
    print(i,end=" ")
    i= i+1
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
1 2 3 4
>>>
```

```
i=1
while (i>5):
    print(i,end=" ")
    i= i+1
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
>>> |
```

❖ Odd number

```
i=1
while (i<10):
    print(i,end=" ")
    i= i+2
```

```
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
1 3 5 7 9
>>> |
```

❖ Square number

```
i=1
while (i<=10):
    print(i*i,end=" ")
    i= i+1
```

```
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
1 4 9 16 25 36 49 64 81 100
>>> |
```

```
i=0
while (i<7):
    i= i+2
    print(i,end=" ")
    i=i+1
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
2 5 8
>>> |
```

```
i=1
while (i<7):
    i= i+1
    print(i+1,end=" ")
    i=i+2
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
3 6
>>> |
```

```
num=1
sum=0
while (num<=10):
    sum=sum+num
    print(sum, end=" ")
    num =num+1
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
1 3 6 10 15 21 28 36 45 55
>>> |
```

❖ Sum of the first integers

```
num=1
sum=0
while (num<=10):
    sum=sum+num
    num =num+1
print("sum of first ten integers:-", sum )
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
sum of first ten integers:- 55
>>> |
```

```

n=1
a=1
while (n<=10):
    print(a, end=' ')
    a=a*2
    n=n+1

```

```

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
1 2 4 8 16 32 64 128 256 512
>>> |

```

❖ Note

- If End = ' ' used and print on next line(\n)

❖ Example

```

n=1
while (n<10):
    n=n+3
    print(n+3, end=' ')
    n=n+1
print("\nn=",n)

```

```

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
7 11 15
n= 13
>>> |

```

```

sum=0
num=1
n=int(input("How many numbers you want to add:"))
while (num<=n):
    sum=sum+num
    num=num+1
print(sum)

```

```

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
How many numbers you want to add:2
3
>>> |

```

❖ Factorial program

```
n=int(input("Enter a number:"))
fact=1
while (n>1):
    n=n-1
    fact=fact*n
print(fact)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
Enter a number:5
120
>>>
===== RESTART: C:/Users/L.Visagan/Desktop/sss.py =====
Enter a number:4
24
>>> |
```

For loop

```
for i in range (1,6):
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
1 2 3 4 5
>>> |
```

```
for i in range (1,10,2):
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
1 3 5 7 9
>>> #Increases by 2 |
```

```
for i in range (10,1,-2):  
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
10 8 6 4 2  
>>> |
```

```
for i in [1, 3, 5, 4, 2]:  
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
1 3 5 4 2  
>>> |
```

```
for i in (1, 3, 5, 4, 2):  
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
1 3 5 4 2  
>>> |
```

Break / Continue

```
for i in "Python-Programming":
    if i=="o":
        break
    print(i, end=' ')
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
P y t h
>>> |
```

```
for i in "Python-Programming":
    if i=="o":
        continue
    print(i, end=' ')
```

```
file edit shell debug options window help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
P y t h o n - P r o g r a m m i n g
>>> |
```

User defined function

பயனர் வரையறை செயலி

- def function name():
- A function must be called otherwise, it will not work.

- செயலி ஒன்று கட்டாயமாக அழைக்கப்படுதல் வேண்டும்.

```
def greetings():  
    print("Congrats")  
greetings()# calling the function name
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
Congrats  
>>> |
```

```
def greetings():  
    print("Congrats")  
greetings()  
greetings()  
greetings()
```

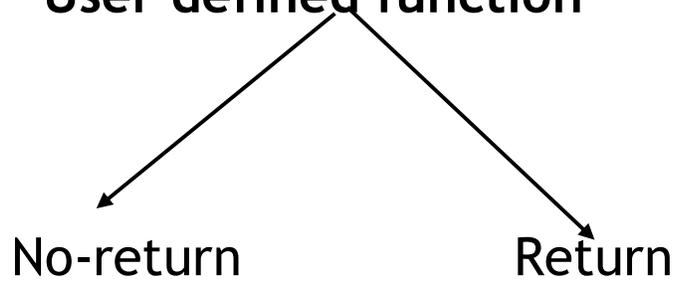
```
file edit shell debug options window help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
Congrats  
Congrats  
Congrats  
>>> |
```

❖ Parameter passing

```
def calc(a,b):  
    print(a+b)  
calc(4,5)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
9  
>>> |
```

User defined function



❖ No-return type

```
def calc(a,b):  
    if a>b:  
        print(a)  
    else:  
        print(b)  
calc(4,5)
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
5  
>>> |
```

❖ Return type

```
def calc(a,b):  
    if a>b:  
        return(a)  
    else:  
        return(b)  
print(calc(4,5))
```

```
file edit shell debug options window help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
5  
>>> |
```

❖ Nested loop

for....
for....

while...
while....

for....
while....

```
for i in range(0,4):  
    print()  
    for j in range (0,3):  
        print( 'x',end=" ")
```

```
File Edit Shell Debug Options Window Help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (In  
tel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
  
x x x  
x x x  
x x x  
x x x  
>>> |
```

```
File Edit Shell Debug Options Window Help
for i in range(0,4):
    for j in range(0,3):
        print(i,end=" ")
    print()

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
0 0 0
1 1 1
2 2 2
3 3 3
>>> |
```

```
for i in range(0,3):
    for j in range(0,4):
        print(j,end=" ")
    print()

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
0 1 2 3
0 1 2 3
0 1 2 3
>>> |
```

```
File Edit Shell Debug Options Window Help
for i in range(0,3):
    for j in range(0,4):
        print(i+j,end=" ")
    print()

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
0 1 2 3
1 2 3 4
2 3 4 5
>>> |
```

```
File Edit Shell Debug Options Window Help
for i in range(1,5):
    for j in range(0,i):
        print(j,end=" ")
    print()

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py
0
0 1
0 1 2
0 1 2 3
>>> |
```

❖ Multiplication table

```
def multiplication (a):  
    for i in range (1,17):  
        print (a,"x",i ,"=", a*i)  
multiplication (5)
```

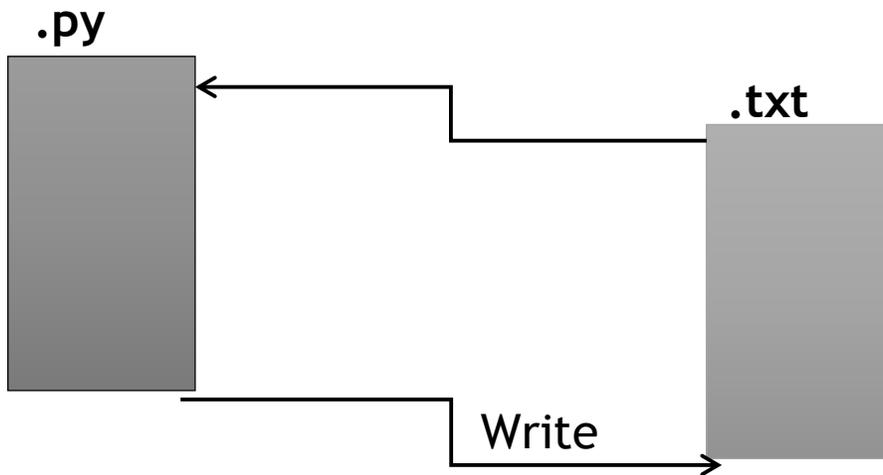
```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Users/L.Visagan/AppData/Local/Programs/Python/Python35-32/abd.py  
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50  
5 x 11 = 55  
5 x 12 = 60  
5 x 13 = 65  
5 x 14 = 70  
5 x 15 = 75  
5 x 16 = 80  
>>>
```

- All the multiplications of 1-12 will visible up to 16 in order.

```
for i in [1,2,3,4,5,6,7,8,9,10,11,12]:
    print ("multiplication table of ",i )
    print(".....")
    for j in range (1,17):
        print (i ,"x" ,j ,"=" ,i*j )
```

```
multiplication table of 1
.....
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
1 x 11 = 11
1 x 12 = 12
1 x 13 = 13
1 x 14 = 14
1 x 15 = 15
1 x 16 = 16
multiplication table of 2
.....
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24
2 x 13 = 26
2 x 14 = 28
2 x 15 = 30
2 x 16 = 32
multiplication table of 3
.....
3 x 1 = 3
3 x 2 = 6
```

Text file handling



R=Read mode
W=write mode
A=append mode

❖ Open function

- `f=open('test.txt')`
`print (f)`
- `open()`: The built in functions takes the name of the file as a parameter and returns a file object.

OR

- `f=open ('test.txt' 'r')`

`print(f)`

- `open()`-இச் செயலி (function) ஒரு பாடக்கோப்பின்(text file) பெயரினை எடுத்து ஒரு கோப்பு இலக்கு பொருளாக (file object) திரும்பலை வழங்குகிறது.

❖ `Readline()`

- `f=open ('test.txt')`

`print (f.readline ())`

- `readline ()`: reads characters from the file until it get to newline and returns the results as as string.
- பாடக்கோப்பிலுள்ள உள்ளடக்கத்தின் புதிய வரியினை பெறும் வரை ஒவ்வொரு வரியுருக்களாக (characters) வாசிக்கின்றது.

❖ Read ()

- `f=open('test.txt')`
`print (f.read ())`

❖ Strip()

- `f=open ('test.txt')`
`line= f.readline ()`
`a =line.strip ()`
`print (a)`
`f.close()`
- Strip returns a copy of the string in which all characters have been stripped from the beginning and the end of the string.
- வரியுருச்சம் ஒன்றின் ஆரம்பத்திலிருந்து இறுதிவரை ஒவ்வொரு வரியுருக்களினையும் ஒன்று சேர்க்கின்றது.

❖ Write () function

- `f=open('test.txt','w')`
`f.write('welcome')`
`f.close ()`
- Replaces the text on test.txt by welcome
- `f=open ('test.txt','w')`
`f.write ('python\n')`
`f.write ('programming')`
`f.close ()`
- ***Output***
python
programming
- `f=open('test.txt','a')`
`f.write ('welcome')`
`f.close ()`

- ***Output***

```
python  
programming welcome
```

- ❖ **Example**

```
f1=open ('input.txt','r')  
f2=open ('output.txt','w')  
for line in f1:  
    data= (line. strip ()).split (“, “)  
    total=float (data [1]) +float (data [2])  
    f2.write (“%75-%4d\n %( data[a], total))  
f1.close ()  
f2.close ()
```

Input.txt

Nimal, 30, 60

Saman, 80, 45

Upali, 100, 80

Output.txt

Nimal-90

Saman -125

Upali-180

❖ Example

```
f1=open ('input.txt','r')
f2=open ('output.txt','w')
line=f1.readline ()
while(line):
    data=line. Strip ()).split (“,”)
    total=float (data [1]) +float (data [2])
    f2.write(‘{},{},{ },{ },\n
format(data[0],data[1],data[2],total))
    line=f1.readline ()
f1.close ( )
f2.close ()
```

Output

```
{ },{ },{ },{ }
```

Input.txt

Nimal, 30, 60

Saman, 80, 45

Upali, 100, 80

Output.txt

Nimal, 30, 60, 90.0

Saman 80, 45,125.0

Upali 100, 80,180.0

Consider the following python program

```
data = [5, 1, 23, 10,-3]
```

```
def fun (a):
```

```
    i, c=1, a[0]
```

```
    while i<len(a):
```

```
        if (a[i]>c):
```

```
            c=a(i)
```

```
            i=i+1
```

```
    return
```

```
print (fun (data))
```

Programming Errors / Bugs

செய்நிரலாக்க வழக்கள்

- Compile time / Syntax error
- Run - time error
- Logical / Semantic error

❖ Compile time / Syntax error

❖ தொகுப்பு நேர/ தொடரியல் வழு

- A syntax error is an error in the syntax of a sequence of characters.
- Syntax errors are detected at compile time.
- A program will not compile until all syntax errors are corrected.
- Shows error message.
- தொடரியல் வழுக்கள் செய்நிரலின் குறியீடுகளில் ஏற்படும் பிழைகளினால் ஏற்படுகின்றன.
- இவ்வழுக்கள் செய்நிரல் தொகுக்கப்படும் போது ஏற்படுகின்றன.

- இவ்வழுக்கள் திருத்தப்படும் வரை செய்நிரல்கள் நிறைவேற்றப்படமாட்டாது.
- இவ்வழுக்கள் வழுச்செய்திகளை காண்பிக்கின்றன.

❖ Example

- x=2
print(x)
- x=2
print(x)
- for i in range (1,10)
print (i)

❖ Run -time error

❖ ஓட்டநேரவழு

- A type of error that occurs during the execution of a program is known as run time error.
- Run time errors are usually more difficult to find and fix than syntax errors.
- Shows error message.
- ஓட்டநேர வழு செய்நிரலானது நிறைவேற்றப்படும் போது உருவாகின்றது.
- ஓட்ட நேர வழுக்கள் ஏனைய வழுக்களை விட கையாள்வதற்கு கடினமானவை.
- இவ்வழுக்கள் வழுச்செய்திகளை காண்பிக்கின்றன.

❖ Example

x=0

y=0

print (x/y)

❖ Logical / Semantic Error

❖ தருக்க / சொற்பொருளியல் வழு

- Unexpected results occur.
- These are valid code the compiler understands but they do not have what programmer intended.
- Does not show error message.
- எதிர்பார்க்கப்படாத வருவிளைவு தோன்றுதல்
- செய்நிரல் தொடரியல் ரீதியில் சரியானதாக இருந்த போதிலும் எதிர்பார்க்கப்பட்ட வருவிளைவு கிடைக்க பெறமாட்டாது.
- இவை வழுச்செய்தியை காண்பிப்பதில்லை.

Escape Sequence / Characters

- `print ("Python\n Programming\n Language")`

In -New line

Output

```
Python
Programming
Language
```

- It -tab space
`print ("Python\t Programming\t language")`

Output

```
Python      Programming      Language
```

- `\'` - Single quote
`print (Vimal\'s address)`

Output

```
Vimal's address
```

- `\"` Double Quote
`print ('CPU \'Central Processing Unit\')`

Output

```
CPU "Central Processing Unit"
```

- `Import math`
`print (sum ((3, 6, 6, 5, 7)))`

Output

```
27
```

- import math
print (abs (-56.0)) #absolute value (Comment)

Output

56.0

- Import math
print (max (5, 8, 3))

Output

8

- Import math
print (min (5, 8, 3))

Output

3

- Import math
print (math. Sin (45))

Output

.tan

.cos

- Import math
print (math.floor (6.567))

Output

6

- Import math
print (math.ceil (2.567))

Output

3

- Import math
print (math.trunc (2.567))
#Truncate (drop decimal digits)

Output

2

- Import math
print (round (2.567))
rounding off

Output

3

- Import math
print ('%2f'%2.567, '{0:.2f}')
Format (2.567)

Output

2.57 2.57

- Import math

```
print ("%2f" % (1/3))
```

Output

0.33

❖ Example 01

```
x=2
```

```
y=5
```

```
for j in range (0, y, x):
```

```
    print (j)
```

```
    print (x+y)
```

```
print ("Done")
```

Output

0

7

2

7

4

7

Done

❖ Example 02

```
ans=0
for i in range (1, 11):
    ans=ans+i*i
    print (i, end=' ')
print (ans)
```

Output

```
1 2 3 4 5 6 7 8 9 10
385
```

❖ Example 03

```
a= ['cat', 'window', 'moon']
For x in a:
    print (x,len(x))
```

Output

```
cat 3
window 6
moon 4
```

❖ Example 04

```
a = [66.25, 333, 333, 1, 1234.5]
print (a.count (333))
```

Output

2

❖ Example 05

```
a= (5, 7, 8, 4)
a.reverse ( )
print (a)
```

Output

[4, 8, 7, 5]

Set

தொடை

- Set is a collection which is unordered and unindexed.
- No duplicate members.
- தொடை என்பது ஒழுங்குபடுத்தப்படாததும் சுட்டிடப்படாததுமான ஒரு தொகுதி ஆகும். அது ஒரே பெறுமதிகளை அனுமதிப்பதில்லை.

- `s={4,6,5,3,1}`
`print (s)`
Output
`{1, 3, 4, 5, 6}`

- `s={}`
`print(s)`
Output
`}`

- `s={3,5,9,1,4}`
`print(s [1])`

ERROR

- `s1={3,5}`
`s2= {1, 6, 5}`
`print (S1= =S2)`
Output
`False`

- `s1={3,5}`
`s2= {3, 5}`
`print (S1= =S2)`
Output
`True`

❖ Note

- **List:** - list is a collection which is ordered and changeable. Allow duplicate members (Mutable)
- **Tuple:** - Tuple is a collection which is ordered and unchangeable. Allow duplicate members. (Immutable)
- **Dictionary:** - Dictionary is a collection which is unordered, changeable and indexed. No duplicate members. (Mutable)
- List என்பது ஒழுங்குபடுத்தப்பட்டதும் மாற்றக்கூடியதுமான ஓர் தொகுதி ஆகும் அது ஒரே பெறுதிகளை அனுமதிக்கின்றது.
- Tuple என்பது ஓர் ஒழுங்குபடுத்தப்பட்டதும் மாற்ற முடியாததுமான ஓர் தொகுதி ஆகும் அது ஒரே பெறுமதிகளை அனுமதிக்கின்றது.
- Dictionary என்பது ஒழுங்குபடுத்தப்படாததும் மாற்றக்கூடியதுமான ஓர் தொகுதி ஆகும் அது ஒரே பெறுமதிகளை அனுமதிப்பதில்லை.

