

# நேஷனல் மேல்நிலைப் பள்ளி, குடியாத்தம்

12 ஆம் வகுப்பு  
கணினி அறிவியல்



பெயர் : \_\_\_\_\_  
வகுப்பு : \_\_\_\_\_  
பிரிவு : \_\_\_\_\_  
தேர்வு எண் : \_\_\_\_\_

1. செயற்கூறு

புத்தக வினாக்கள்:

- ஒரு குறிப்பிட்ட செயலை செய்தற்காக பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே அ) துணை நிரல்கள் ஆகும். இ) Pseudo குறிமுறை ஈ) தொகுதிகள்
- பின்வரும் எந்த அலகு ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்பட்டுள்ளது? அ) துணை நிரல்கள் ஆகும். இ) கோப்புக்கள் ஈ) தொகுதிகள்
- பின்வரும் எது தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும்? அ) துணை நிரல்கள் ஆகும். இ) வரையறைகள் ஈ) தொகுதிகள்
- செயற்கூறு வரையறையில் உள்ள மாறிகள் எவ்வாறு அழைக்கப்படுகிறது? அ) துணைநிரல்கள் ஆகும். இ) அளபுருக்கள் ஈ) செயலுருபு
- செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் எவ்வாறு அழைக்கப்படுகிறது? அ) செயலுருபுகள் ஆகும். இ) துணை நிரல்கள் ஈ) செயற்கூறு
- தரவுவகை குறிப்பு எழுதும்போது, எது கட்டயமாகிறது? அ) {} ஆகும். இ) [] ஈ) <>

2 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

- துணைநிரல் என்றால் என்ன?  
ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக மீண்டும் மீண்டும் பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே துணை நிரலாக்கம் எனப்படும்.
- நிரலாக்க மொழியைப் பொறுத்து செயற்கூறுவை வரையறுக்கவும்.  
செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும். இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும்.
- $x := (78)$  இதன் மூலம் அறிவது என்ன?  
78 என்ற மதிப்பு 'x' என்ற பெயருடன் பிணைக்கின்றது.
- பின்வருவனவற்றுள் எது சாதாரண செயற்கூறு வரையறை மற்றும் எது தற்சுழற்சி செயற்கூறு வரையறை  
i) let rec sum x y:  
return x + y

விடை:

தற்சுழற்சி செயற்கூறு

ii) let disp:

print 'welcome'

விடை:

சாதாரண செயற்கூறு

iii) let rec sum num:

if (num!=0) then return num + sum (num-1)

else

return num

விடை:

தற்சுழற்சி செயற்கூறு

**5 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. செயலுருபுகள் என்றால் என்ன?

- அ) தரவு வகை கில்லாத அளபுருக்கள்
- ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் விவரி.

அ) தரவு வகை கில்லாத அளபுருக்கள்:

தரவு வகை கில்லாத அளபுருக்கு ஓர் எடுத்துக்காட்டை காணலாம்.

```
(requires: b >= 0)
(returns: a to the power of b)
let rec pow a b :=
  if b = 0 then 1
  else a * pow b (a-1)
```

- மேலே உள்ள எடுத்துக்காட்டில் எந்த தரவினத்தையும் குறிப்பிடவில்லை.
- if கோவை முழுவதுமே 'int' தரவு வகைக் கொண்டிருக்கும். if கோவையின் தரவு வகை int ஆக இருப்பதால் செயற்கூறின் திருப்பி அனுப்பும் மதிப்பும் int ஆக இருக்கும்.
- 'b' ன் மதிப்பு சுழியத்தோடு = செயற்கூறியுடன் ஒப்பீடு செய்யப்படுகிறது. அதனால் 'b' ன் தரவு வகையையும் 'int' ஆகும்.
- செயற்கூறியுடன் 'a' ன் மதிப்பு மற்றொரு கோவையோடு பெருக்குத்தொகையைக் கணக்கிடுவதால், 'a' ன் வகையும் 'int' ஆகும்.

ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள்:

```
(requires: b > 0)
(returns: a to the power of b)
let rec pow (a: int) (b: int) : int :=
  if b = 0 then 1
  else a * pow b (a-1)
```

- மேலே உள்ள எடுத்துக்காட்டில் a மற்றும் b ன் தரவு வகை குறிப்பிடப்பட்டுள்ளது.
- அதேபோல் திருப்பு அனுப்பும் தரவு வகையும் குறிப்பிடப்பட்டுள்ளது.
- தரவு வகை குறிப்பு எழுதும்போது அடைப்புக்குறி ( ) அவசியமாகும்.
- தரவு வகையை வெளிப்படையாக குறிப்பிடுவது, பிழைச் செய்தியைத் திருத்தம் செய்வதற்கு பயனுள்ளதாக இருக்கும்.

2. பின்வரும் நிரலில்

```
let rec gcd a b :=
  if b <> 0 then gcd b (a mod b) else return a
```

அ) செயற்கூறுவின் பெயர்

gcd

ஆ) தற்குழற்சி செயற்கூறு கூற்று

let rec gcd a b :=

இ) அளபுருக்கள் கொண்ட மாறியின் பெயர்

a மற்றும் b

ஈ) செயற்கூறுவை தற்கழற்சிக்கு அழைக்கும் கூற்று

gcd b (a mod b)

2 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

1. செயற்கூறு வரையறையின் தொடரியலை எழுதுக.

செயற்கூறு வரையறையின் தொடரியல்:

*let rec fn a1 a2 ..... an: = k*

2. தற்கழற்சி செயற்கூறு என்றால் என்ன?

தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறு வரையறைக்கு தற்கழற்சி செயற்கூறு என்று பெயர்.

3. வரையறு: அளபுருக்கள், செயலுருபுகள்

அளபுருக்கள்:

அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும்.

செயலுருபுகள்:

செயலுருபுகள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.



**பாடம் 2 தரவு அருவமாக்கம்**

**புத்தக வினாக்கள்:**

1. பின்வரும் எந்த செயற்கூறு அருவமாக்கம் தரவு வகையை உருவமைக்கப் பயன்படுகிறது?

**A) Constructors**    ஆ) Destructors    இ) recursive    ஈ) Nested

2. பின்வரும் எந்த செயற்கூறு தரவு வகையில் இருந்து தகவல்களை மீட்டெடுக்கும்?

அ) Constructors    **ஆ) Selectors**    இ) recursive    ஈ) Nested

**2 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. தரவு அருவமாக்கம் வகை என்றால் என்ன?

தரவு அருவமாக்கம் என்பது தேவையற்ற விவரங்களை மறைத்து அவசியமானவற்றை மட்டும் வழங்கும் செயல்முறை ஆகும்.

2. ஆக்கிகள் மற்றும் செலக்டர்கள் வேறுபாடு தருக?

- ஆக்கி செயற்கூறுகள், அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது.
- செலக்டர் செயற்கூறுகள், தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.

**3 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. பின்வருவனவற்றில் எது constructors மற்றும் selectors என்று அடையாளம் காண்க.

- (a) N1 = number()                      (b) accetnum (n1)                      (c) displaynum (n1)  
 (d) eval (a/b)                              (e) x,y = makeslope (m), makeslope (n)  
 (f) display ()

- |  |   |             |
|--|---|-------------|
| (a) N1 = number()                      | - | Constructor |
| (b) accetnum (n1)                      | - | Selector    |
| (c) displaynum (n1)                    | - | Selector    |
| (d) eval (a/b)                         | - | Selector    |
| (e) x,y = makeslope (m), makeslope (n) | - | Constructor |
| (f) display ()                         | - | Selector    |

**5 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. தரவு அருவமாக்கம் எவ்வாறு செயல்படுத்துவாய்? எடுத்துக்காட்டுடன் விளக்குக.

- தரவு அருவமாக்கத்தினை செயல்படுத்த ஆக்கிகள் மற்றும் செலக்டர்கள் என்ற இரண்டு செயற்கூறுகள் உருவாக்கப்பட வேண்டும்.
- ஆக்கி செயற்கூறுகள், அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது.
- செலக்டர் செயற்கூறுகள், தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.

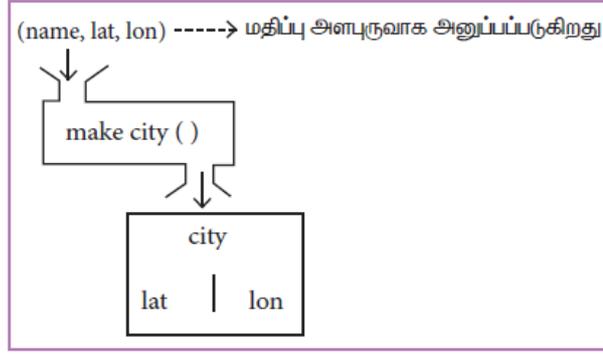
எடுத்துக்காட்டாக, city என்ற ஒரு அருவமாக்க தரவு வகை உள்ளது என வைத்துக் கொள்வோம். city என்ற பொருள் நகரத்தின் பெயர், அட்சரேகை மற்றும் தீர்க்கரேகை பற்றிய தகவல்களை சேமித்திருக்கும் city என்ற பொருளை உருவாக்க பின்வரும் செயற்கூற்றினை பயன்படுத்தலாம்.

city = makecity (name, lat, lon)

city என்ற பொருளின் தகவல்களை பெறுவதற்கு பின்வரும் செயற்கூறுகளைப் பயன்படுத்தலாம்.

- getname (city)
- getlat (city)
- getlon (city)

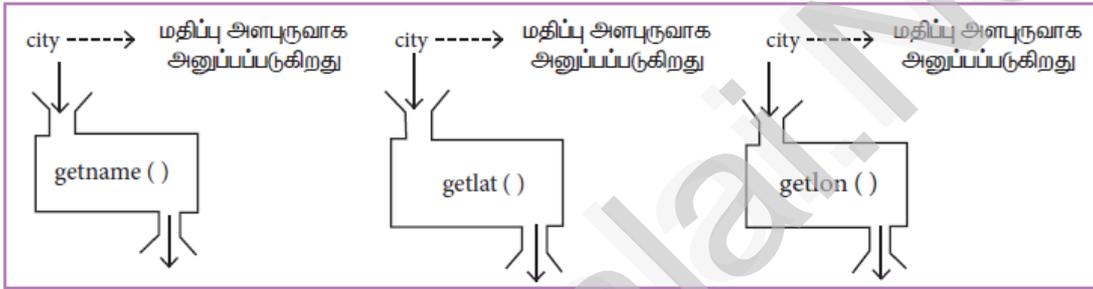
இங்கு makecity (name, lat, lon) என்ற ஆக்கி city எனும் பொருளை உருவாக்குகிறது.



செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது. மேலே காணும் குறிமுறையில்,

- getname (city)
- getlat (city)
- getlon(city)

என்பவை city எனும் பொருளிலிருந்து தகவல்களை பெற்று தரும் செலக்டர் செயற்கூறுகளாகும்.



### கூடுதல் வினாக்கள்:

1. விவரங்களை மறைத்து அவசியமானவற்றை மட்டும் வழங்கும் செயல்முறையை \_\_\_\_\_ என்கிறோம்  
அ) Constructors    ஆ) Selectors    **இ) அருவமாக்கம்**    ஈ) Nested
2. பின்வரும் எது constructors என்று அடையாளம் காண்க.  
அ) display()    **ஆ) n1=number()**    இ) goto    ஈ) for
3. பின்வரும் எது selectors என்று அடையாளம் காண்க.  
**அ) display()**    ஆ) n1=number()    இ) goto    ஈ) for

**பாடம் 3 வரையெல்லை**

**புத்தக வினாக்கள்:**

1. பின்வருவனவற்றுள் எது நிரலின் ஒரு பகுதியின் அணுகியல்பை மற்றொரு பகுதிக்கு குறிப்பதாகும்?  
**அ) வரையெல்லை**      ஆ) நினைவகம்      இ) முகவரி      ஈ) அணுகுமுறை
2. மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையை என்னவென்று அழைக்கப்படும்?  
அ) வரையெல்லை      **ஆ) மேப்பிங்**      இ) பின் பிணைத்தல்      ஈ) முன் பிணைத்தல்
3. பின்வருவனவற்றுள் எது நிரலாக்க மொழியில் மாறியையும் பொருளையும் மேப் செய்யப் பயன்படுகிறது  
அ) ::      ஆ) :=      **இ) =**      ஈ) ==
4. எது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம் ஆகும்  
அ) வரையெல்லை      ஆ) மேப்பிங்      இ) பிணைத்தல்      **ஈ) Namespaces**
5. எந்த வரையெல்லை நடப்பு செயற்கூறில் வரையறுக்கப்படும் மாறிகளைக் குறிக்கும்?  
**அ) உள்ளமை வரையெல்லை**      ஆ) முழுதளாவிய வரையெல்லை  
இ) தொகுதி வரையெல்லை      ஈ) செயற்கூறு வரையெல்லை

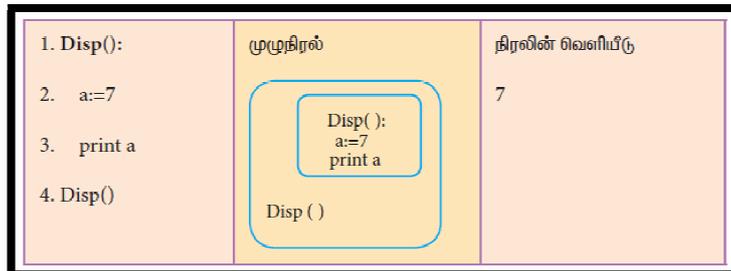
**2 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. வரையெல்லை என்றால் என்ன?  
வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற்கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும்.
2. மாறிகளுக்கு எதற்காக வரையெல்லை பயன்படுத்தப்பட வேண்டும்? காரணம் கூறுக.  
நிரலில் வரையறுக்கப்பட்ட ஒவ்வொரு மாறியும் முழுதளாவிய வரையெல்லைக் கொண்டுள்ளன. ஒருமுறை வரையறுக்கப்பட்டால், நிரலின் ஒவ்வொரு பகுதியும் அந்த மாறியை அணுக முடியும். ஆனால், ஒரே ஒரு வரையறைக்குள் மாறிகளின் வரையெல்லை உட்படுத்திவது சிறந்த வழிமுறை ஆகும். இதில் எதிர்பாராத விதமாக செயற்கூறுக்கு உள்ளே உள்ள மாறிகளில் ஏற்படும் மாற்றங்கள் செயற்கூறுவுக்கு வெளியே எந்த மாற்றத்தையும் ஏற்படுத்தாது.
3. மேப்பிங் என்றால் என்ன?  
மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையே மேப்பிங் எனப்படும்.
4. Namespaces சிறுகுறிப்பு வரைக.  
Namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான கொள்கலனாகும்.

**3 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

1. உள்ளமை வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
  - உள்ளமை வரையெல்லை, நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும்.
  - செயற்கூறு, எப்பொழுதும் மாறியின் பெயரை முதலில் அதன் உள்ளமை வரையெல்லையில் பார்வையிடும்.
  - அந்த வரையெல்லையில் இல்லையென்றால் மட்டுமே வெளி வரையெல்லையில் சோதிக்கும்.

**எடுத்துக்காட்டு:**



2. முழுதளாவிய வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?

- நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும்.
- அதாவது, முழுதளாவிய மாறிகளை நிரலின் அனைத்து செயற்கூறுகளுக்கு உட்புறமும், வெளிப்புறமும் அணுக முடியும்.

எடுத்துக்காட்டு:

1. a:=10	முழுநிரல்	நிரலின் வெளியீடு
2. Disp():		7
3. a:=7		10
4. print a		
5. Disp()		
6. print a		

3. அடைக்கப்பட்ட வரையெல்லை எடுத்துக்காட்டுடன் விவரி?

- மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால், அச்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும்.
- இதுவே அடைக்கப்பட்ட வரையெல்லை எனப்படும்.
- நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும்பொழுது அது முதலில் உள்ளமை வரையெல்லையின் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும்.

எடுத்துக்காட்டு:

1. Disp():	முழுநிரல்	நிரலின் வெளியீடு
2. a:=10		10
3. Disp1():		10
4. print a		
5. Disp1()		
6. print a		
7. Disp()		

4. பின்வரும் போலிக் (pseudo) குறிமுறையில் மாறிகளின் வரையெல்லைக் கண்டறிந்து வெளியீட்டை எழுதுக.

output:

color:= Red

mycolor():

b:=Blue

myfavcolor():

g:=Green

print, b,g

myfavcolor():

printcolor, b

mycolor()

print color

வெளியீடு:

Red Blue Green  
Red Blue  
Red

colour	-	முழுதளாவிய வரையெல்லை
b	-	இணைக்கப்பட்ட வரையெல்லை
a	-	உள்ளமை வரையெல்லை

5 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

1. மாறிகளின் வரையெல்லைகளின் வகைகளை விளக்குக. (அல்லது) LEGB விதியை எடுத்துக்காட்டுடன் விளக்குக.

LEGB விதி வரையெல்லை தேடப்பட வேண்டிய வரிசையை தீர்மானிக்கப் பயன்படுகிறது. வரையெல்லைகள் பின்வருமாறு படிநிலை முறையில் பட்டியலிடப்பட்டுள்ளன.

உள்ளமை Local (L)	செயற்கூறு/இனக்குழுவிற்கு உள்ளே வரையறுக்கப்பட்டவை
இணைக்கப்பட்ட Enclosed (E)	பின்னலான செயற்கூறுகளுக்குள் வரையறுக்கப்பட்டவை
முழுதளாவிய Global (G)	மேல்நிலையில் வரையறுக்கப்பட்டவை
உள்ளிணைந்த Built in (B)	உள்ளிணைந்த செயற்கூறுகளில் உள்ள முன்னரே வரையறுக்கப்பட்ட பெயர்களாகும்.

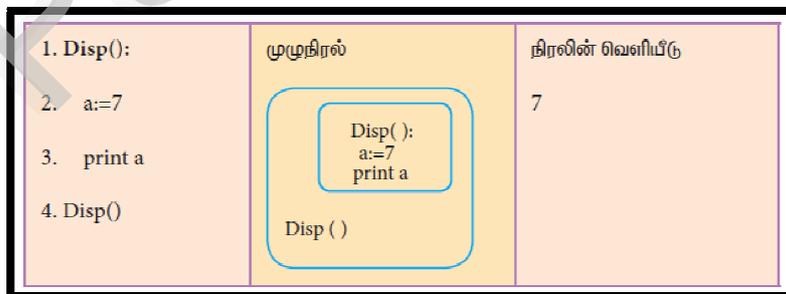
மாறியின் வரையெல்லை வகைகள்:

4 வகையான வரையெல்லைகள் உள்ளன. அவை முறையே,

உள்ளமை வரையெல்லை:

- உள்ளமை வரையெல்லை, நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும்.
- செயற்கூறு, எப்பொழுதும் மாறியின் பெயரை முதலில் அதன் உள்ளமை வரையெல்லையில் பார்வையிடும்.
- அந்த வரையெல்லையில் இல்லையென்றால் மட்டுமே வெளி வரையெல்லையில் சோதிக்கும்.

எடுத்துக்காட்டு:



முழுதளாவிய வரையெல்லை:

- நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும்.
- அதாவது, முழுதளாவிய மாறிகளை நிரலின் அனைத்து செயற்கூறுகளுக்கு உட்புறமும், வெளிப்புறமும் அணுக முடியும்.

**எடுத்துக்காட்டு:**

	முழுநிரல்	நிரலின் வெளியீடு
1. a:=10		
2. Disp():		7
3. a:=7		10
4. print a		
5. Disp()		
6. print a		

**அடைக்கப்பட்ட வரையெல்லை:**

- மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால், அச்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும்.
- இதுவே அடைக்கப்பட்ட வரையெல்லை எனப்படும்.
- நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும்பொழுது அது முதலில் உள்ளமை வரையெல்லையின் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும்.

**எடுத்துக்காட்டு:**

	முழுநிரல்	நிரலின் வெளியீடு
1. Disp():		
2. a:=10		10
3. Disp1():		10
4. print a		
5. Disp1()		
6. print a		
7. Disp()		

**உள்ளிணைந்த வரையெல்லை:**

- நிரல்பெயர்ப்பி அல்லது தொகுப்பானை தொடங்கும் பொழுது உள்ளிணைந்த வரையெல்லையானது நிரல் வரையெல்லையெனில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும்.
- நிரலாக்க மொழியின் நூலக செயற்கூறினுள் வரையறுக்கப்பட்ட மாறி அல்லது தொகுதி உள்ளிணைந்த வரையெல்லைக் கொண்டுள்ளது.
- இவைகள், நூலக கோப்புகள் நிரலில் செயல்பட தொடங்கியவுடன் இறக்கப்படும்.

	முழுநிரல்	நிரலின் வெளியீடு
1. Disp():		
2. a:=10		10
3. Disp1():		10
4. print a		
5. Disp1()		
6. print a		
7. Disp()		

**பாடம் 4 நெறிமுறையின் யுக்திகள்**

**புத்தக வினாக்கள்:**

- எந்த சொல் பெர்ஷிய கணித மேதை அபு ஜா.:பர் முகமது ஐமுசா அல் கௌவாரிஸ்மி பெயரில் இருந்து வந்தது?  
அ) Flowchart      ஆ) Flow      இ) **Algorithm**      ஈ) Syntax
- பின்வரும் வரிசையாக்க நெறிமுறையில், எந்த நெறிமுறைக்கு குறைந்த எண்ணிக்கையிலான இடமாற்றம் தேவைப்படும்?  
அ) குமிழி      ஆ) விரைவு      இ) ஒன்றிணைந்த      ஈ) **தேர்ந்தெடுப்பு**
- பின்வரும் வரிசையாக்க நெறிமுறையில் எது மிகவும் குறைவான மோசமான சிக்கல்தன்மையை உடையது?  
அ) குமிழி      ஆ) விரைவு      இ) **ஒன்றிணைந்த**      ஈ) தேர்ந்தெடுப்பு
- பின்வருவனவற்றுள் எது நிலையான வரிசையாக்க நெறிமுறை அல்ல?  
அ) செருகதல்      ஆ) **தேர்ந்தெடுப்பு**      இ) குமிழி      ஈ) ஒன்றிணைந்த

**2 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

- நெறிமுறை என்றால் என்ன?  
ஒரு குறிப்பிட்ட செயலை நிறைவேற்றுவதற்கான வரையறுக்கப்பட்ட கட்டளைகளின் தொகுப்பு நெறிமுறையாகும்.
- வரிசையாக்கம் என்றால் என்ன?  
ஒரு குறிப்பிட்ட முறையில் வழிமுறைகளைப் பயன்படுத்தி உருப்படிக்களை வரிசைப்படுத்தும் செயல்முறையே வரிசையாக்கம் எனப்படும்.
- தேடல் என்றால் என்ன? அதன் வகைகளை எழுதுக.  
தரவு கட்டமைப்பில் உள்ள ஒரு உருப்படியை தேடும் செயல்முறையே தேடல் எனப்படும். தேடல் இரண்டு வகைப்படும். அவை முறையே,
  - வரிசைமுறைத் தேடல்
  - இருமத் தேடல்

**3 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

- நெறிமுறையின் பண்பியல்புகளைப் பட்டியலிடுக.
  - உள்ளீடு
  - வெளியீடு
  - எல்லையற்றது
  - வரையறுத்தல்
  - செயல்தன்மை
  - உண்மைத்தன்மை
  - எளிமை
  - குழப்பமற்றது
  - செயலாக்கம்
  - அடக்கமானது
  - சார்பற்றது

**5 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)**

**1. நெறிமுறையின் பண்பியல்புகளை விவரி.**

உள்ளீடு	பூஜ்ஜியம் அல்லது அதிக எண்ணிக்கையில் வழங்கப்பட வேண்டும்.
வெளியீடு	குறைந்தபட்சம் ஒன்றாவது உருவாக்கப்பட வேண்டும்.
எல்லையற்றது	வரையறுக்கப்பட்ட எண்ணிக்கையிலான படிநிலைகளில் நெறிமுறை நிறுத்தப்பட வேண்டும்.
வரையறுத்தல்	அனைத்து செயல்பாடுகளும் நன்றாக வரையறுக்கப்பட வேண்டும்.
செயல்தன்மை	ஒவ்வொரு கட்டளைகளும் திறம்பட நடத்தப்பட வேண்டும்.
உண்மைத்தன்மை	நெறிமுறைகள் பிழை இல்லாததாக இருக்க வேண்டும்.
எளிமை	செயல்படுத்துவதற்கு மிக எளிதாக இருக்க வேண்டும்.
குழப்பமற்றது	நெறிமுறையானது தெளிவாகவும் குழப்பமற்றதாகவும் இருத்தல் வேண்டும்.
செயலாக்கம்	கிடைக்கும் வளங்களை வைத்து செயலாக்க வல்லது.
அடக்கமானது	நெறிமுறை பொதுவானதாக இருக்க வேண்டும். நிரலாக்க மொழி மற்றும் இயக்க அமைப்பை சாராமல் இருக்க வேண்டும்.
சார்பற்றது	எந்த நிரலாக்க குறிமுறையை சாராமல் இருக்க வேண்டும்.

**2. வரிசைமுறை தேடல் நெறிமுறையை விவாதிக்கவும்.**

வரிசைமுறைத் தேடல் அல்லது தொடர் தேடல் என்பது பட்டியலில் விரும்பும் உறுப்பைக் கண்டுபிடிக்கும் வரை அல்லது பட்டியல் முடியும் வரை வரிசையிலுள்ள ஒவ்வொரு உறுப்புகளையும் சரிபார்த்து, குறிப்பிட்ட மதிப்பைப் பட்டியலில் கண்டுபிடிக்கும் வழிமுறையாகும்.

**போலிக்குறிமுறை:**

- for மடக்கினைப் பயன்படுத்தி அணியில் பயணித்தல்
- ஒவ்வொரு சுழற்சியிலும், இலக்கு மதிப்பை தற்போதைய மதிப்புடன் ஒப்பிடவும்.
  - மதிப்புகள் பொருத்தமாக இருந்தால் அணியின் தற்போதைய சுட்டெண்ணைத் திருப்பி அனுப்பும்.
  - மதிப்புகள் பொருந்தாவிட்டால் அணியில் அடுத்துள்ள உறுப்புக்கு சென்று விரும்பும்.
- பொருத்தம் எதுவும் இல்லையெனில் -1 மதிப்பைத் திருப்பியனுப்பும்.

கீழே கொடுக்கப்பட்டுள்ள அணியின் 25 என்ற எண்ணைத் தேடுவதற்கு, வரிசைமுறைத் தேடலானது படிப்படியாக தொடர் வரிசையில் கொடுக்கப்பட்ட அணியின் முதல் உறுப்பிலிருந்து தேடலைத் தொடங்கும்.

தேடப்படும் உறுப்பு கண்டுபிடிக்கப்பட்டால் அந்த சுட்டெண் திரும்ப அனுப்பப்படும். இல்லையெனில், அணியின் இறுதி சுட்டெண் வரை தேடல் தொடரும். இதில் எண் 25 சுட்டெண் 3 இல் காணப்படுகிறது.

அணியின் சுட்டு	0	1	2	3	4
மதிப்புகள்	10	12	20	25	30

### 3. இருமத் தேடல் என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.

இருமத்தேடலை பாதி இடைவெளித் தேடல் நெறிமுறை என்றும் அழைக்கலாம். வரிசைப்படுத்தப்பட்ட அணிக்குள் இலக்கு மதிப்பின் இருப்பிடத்தைக் கண்டுபிடிக்கிறது. பிரித்து-கைப்பற்றுதல் நெறிமுறையைப் போல் இருமத் தேடலைச் செய்து மடக்கை நேரத்தில் நிறைவேற்றப்படும்.

#### இருமத் தேடலுக்கான போலிக் குறிமுறை:

1) மைய உறுப்பிலிருந்து தொடங்கவும்.

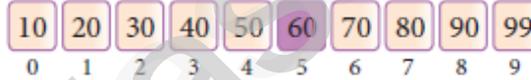
- இலக்கு மதிப்பும் அணியின் மைய உறுப்பும் நிகர் எனில் (அதாவது, மைய இலக்கு = உறுப்புகளின் எண்ணிக்கை/2) மைய உறுப்பின் சுட்டெண்ணைத் திருப்பி அனுப்பும்.
- நிகரில்லை என்றால், மைய உறுப்பை மதிப்போடு ஒப்பிடவும்.
- மைய சுட்டெண்ணிலுள்ள எண் இலக்கு மதிப்பை விட பெரியது எனில், மைய சுட்டெண்ணிற்கு வலப்புறம் உள்ள உறுப்புகளைத் தேர்ந்தெடுத்து படிநிலை -1 லிருந்து தொடங்கவும்.
- மைய சுட்டெண்ணிலுள்ள எண் இலக்கு மதிப்பை விட சிறியது எனில் மைய சுட்டெண்ணிற்கு இடப்புறம் உள்ள உறுப்புகளைத் தேர்ந்தெடுத்து படிநிலை -1 லிருந்து தொடங்கவும்.

2) பொருத்தமான தேடல் கண்டுபிடிக்கப்பட்டால், பொருந்திய உறுப்பின் சுட்டெண்ணைத் திருப்பி அனுப்பும்.

3) பொருத்தம் இல்லையெனில் -1 என்ற மதிப்பை திருப்பி அனுப்பும் அல்லது தேடல் நிறைவேற்றப்படவில்லை என்ற தகவலை அறிவிக்கவும்.

#### இருமத் தேடல் இயங்கும் கோட்பாடுகள்:

இருமத்தேடலில் பயன்படும் அணி வரிசையாக்கம் செய்யப்பட்ட அணியாகியிருக்க வேண்டும். இருமத் தேடலை பயன்படுத்தி மதிப்பு 60-ன் இருப்பிடத்தைத் தேடுவதாக எடுத்துக் கொள்வோம்.



முதலில் நாம் அணியின் மைய உறுப்பை  $mid = low + (high - low)/2$  என்ற வாய்ப்பாட்டைப் பயன்படுத்தி தீர்மானிக்க வேண்டும்.

இங்கு,  $0 + (9 - 0)/2 = 4$  (4.5 யின் முழு மதிப்பு எடுத்துக் கொள்ளவும்.) அதனால் அணியின் மையம் 4 ஆகும்.



இப்பொழுது நாம் 4-ம் சுட்டெண் இருப்பிடத்தில் சேமிக்கப்பட்ட மதிப்போடு தேடப்படும் மதிப்பை (அதாவது 60) ஒப்பிடு செய்வோம். 4-ம் சுட்டெண் இருப்பிடத்தில் உள்ள மதிப்பான 50 என்பது, இது தேடப்படும் மதிப்பு கிடையாது. தேடப்படும் மதிப்பானது 50-விட அதிகமாக இருப்பதால்

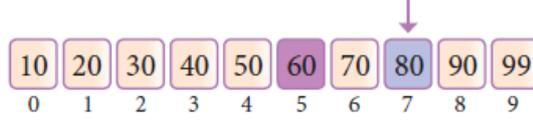


low மதிப்பை  $m+1$  என மாற்றி புதிய mid மதிப்பை மறுபடியும் கண்டுபிடிக்க வேண்டும்.

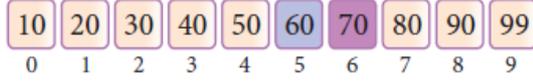
$$low = mid + 1$$

$$mid = low + (high - low)/2$$

இப்பொழுது நமது mid மதிப்பு 7 ஆகும். நாம் இருப்பிடம் 7-ல் சேமிக்கப்பட்ட மதிப்பை இலக்கு மதிப்போடு (அதாவது 60) ஒப்பிடுவோம்.



இருப்பிடம் 7ல் சேமிக்கப்பட்ட மதிப்பு தேடப்படும் மதிப்பு கிடையாது. மாறாக, நாம் தேடுவதை விட அதிகமான மதிப்பாக இருக்கிறது. எனவே, தேடப்படும் மதிப்பு இந்த இருப்பிடத்தை விட குறைவான பகுதியில் இருக்க வேண்டும்.

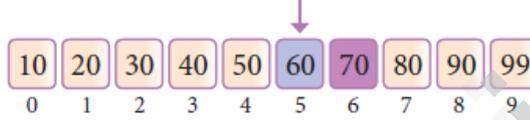


எனவே, நாம் mid மதிப்பைத் திரும்பவும் கணக்கீடு செய்ய வேண்டும்.

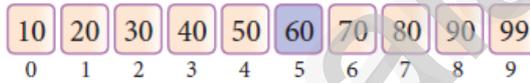
high = mid - 1

mid = low + (high - low)/2

தற்போது mid மதிப்பு 5 ஆகும்.



நாம் இருப்பிடம் 5ல் உள்ள சேமிக்கப்பட்ட மதிப்பை இலக்கு மதிப்போடு ஒப்பீடு செய்வோம். இது ஒரு சரியான பொருத்தமாகும்.



இலக்கு மதிப்பு 60, இருப்பிடம் 5ல் சேமிக்கப்பட்டுள்ளது என்ற நாம் முடிவு செய்கிறோம். எடுத்துக்காட்டாக, இலக்கு மதிப்பு 95 எனில், இந்த செயல்முறை -1 என்ற மதிப்பைத் திருப்பி அனுப்பும்.

#### 4. குமிழி வரிசையாக்க நெறிமுறையை எடுத்துக்காட்டுடன் விளக்குக.

- குமிழி வரிசையாக்கம் ஒரு எளிமையான வரிசையாக்க நெறிமுறை ஆகும்.
- வரிசைப்படுத்தப்பட்ட பட்டியலின் படிநிலைகளை மீண்டும் மீண்டும் செய்து, ஒவ்வொரு ஜோடி அருகிலுள்ள உருப்படிகளை ஒப்பீடு செய்து, வரிசையாக்கம் செய்யப்படாத வரிசை எனில் அவற்றை இடமாற்றம் செய்யும்.
- இடமாற்றம் தேவைப்படும் வரை அவை மீண்டும் மீண்டும் பட்டியலிடப்படும். இது பட்டியல் வரிசையாக்கம் செய்யப்பட்டுள்ளது என்பதைக் குறிக்கும். இந்த ஒப்பீடு வரிசையாக்கம் நெறிமுறையில் பட்டியலின் மேல் பகுதியில் குமிழியைப் போல் சிறிய உறுப்புகளை அமைக்கும் முறையினால் இதற்கு இந்த பெயரிடப்பட்டது.
- இந்த நெறிமுறை எளிதானதாக இருந்த போதிலும், இது மிகவும் மெதுவானது மற்றும் செருகும் வரிசையாக்கத்தோடு ஒப்பீடு செய்யும் போது இது சாத்தியமற்றதாகும்.
- n உறுப்புகளை கொண்ட அணியை கருதிக் கொள்ளவும். இடமாற்ற செயல்முறை மதிப்புகளை இடமாற்றம் செய்யும்.





5. பைத்தான் அறிமுகம் - மாறிகள் மற்றும் செயற்குறிகள்

1 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

- பைத்தானை உருவாக்கியவர் யார்?  
அ)ரிட்சீ ஆ)கைடோ வான் ரோஷம் இ)பில்கேட்ஸ் ஈ)சுந்தர் பிச்சை
- இவற்றுள் எது தூண்டு குறி நிரல் பெயர்ப்பி கட்டளைகளை ஏற்று கொள்ள தயார் நிலையில் இருப்பதை குறிக்கிறது?  
அ)>>> ஆ) <<< இ) # ஈ) <<
- பின்வரும் எந்த சாவி சேர்மானம் ஓர் புதிய பைத்தான் நிரலை உருவாக்கப் பயன்படுகிறது?  
அ) Ctrl + C ஆ) Ctrl + F இ) Ctrl + B ஈ) Ctrl + N
- பின்வரும் எந்த குறியீடு பைத்தான் நிரலில் குறிப்புகளை உள்ளீடு செய்யப் பயன்படுகிறது?  
அ) # ஆ) & இ) @ ஈ) \$
- எந்த குறி ஒன்றுக்கு மேற்பட்ட உறுப்புகளை ஒற்றை வரியில் அச்சிடும்?  
அ)அரைப்புள்ளி ஆ)டாலர் இ)காற்புள்ளி ஈ)முக்காற்புள்ளி
- பின்வரும் எது வில்லைகள் கிடையாது?  
அ)நிரல்பெயர்ப்பி ஆ)குறிப்பெயர்கள் இ)சிறப்புச்சொற்கள் ஈ)செயற்குறிகள்
- பின்வருவனவற்றில் எது பைத்தான் சிறப்புச் சொல் கிடையாது?  
அ) break ஆ) while இ) continue ஈ) operators
- எந்த செயற்குறியை ஒப்பீடு செயற்குறி என்று அழைக்கப்படுகிறது?  
அ)கணக்கீடு ஆ)தொடர்புடைய இ)தருக்க ஈ)மதிப்பிருத்தல்
- பின்வருவனவற்றில் எது தருக்க செயற்குறி கிடையாது?  
அ) and ஆ) or இ) not ஈ) like
- எந்த செயற்குறி நிபந்தனை செயற்குறி என்று அழைக்கப்படுகிறது?  
அ)மும்ம செயற்குறி ஆ)தொடர்புடைய இ)தருக்க ஈ)மதிப்பிருத்தல்

2 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

- பைத்தான் நிரலினை சோதிக்க எத்தனை வகை முறைமைகள் உள்ளன?

பைத்தான் நிரலினை சோதிக்க இரண்டு வகை முறைமைகள் உள்ளன. அவை

- ஊடாகும் முறைமை
- ஸ்கிரிப்ட் முறைமை

- வில்லைகள் பற்றி சிறுகுறிப்பு வரைக.

பைத்தான் நிரலின் இடம்பெறும் வரிகளை அடிப்படை சொற்களாக பிரிக்கிறது. இந்த கூறுகள் வில்லைகள் எனப்படும். வில்லைகள் பிரிப்பதற்கு வெற்று இடைவெளி கண்டிப்பாக இருத்தல் வேண்டும். பொதுவாக வில்லைகளின் வகைகளாவன,

- 1) குறிப்பெயர்கள்
- 2) சிறப்புச்சொற்கள்
- 3) செயற்குறிகள்
- 4) வரம்புக்குறி
- 5) நிலை உரு

- பைத்தானில் உள்ள பல்வேறு செயற்குறிகள் யாவை?

பைத்தானில் உள்ள பல்வேறு செயற்குறிகள் பின்வருமாறு:

- ❖ கணித செயற்குறிகள்
- ❖ ஒப்பீடு செயற்குறிகள்
- ❖ தருக்க செயற்குறிகள்
- ❖ மதிப்பிடுத்து செயற்குறிகள்
- ❖ நிபந்தனை செயற்குறிகள்

4. நிலை உருகள் என்றால் என்ன? நிலை உருகள் வகைகள் யாவை?

நிலை உருகள் என்பது மாறிகள் அல்லது மாறிலிகளுக்கு வழங்கப்படும் மூல தரவாகும். பைத்தானில் பல்வேறு வகையான நிலை உருகள் உள்ளன.

- 1) எண்கள்
- 2) சரம்
- 3) பூலியன்

3 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

1. கணித செயற்குறிகள் பற்றி குறிப்பு வரைக. எடுத்துக்காட்டு தருக.

கணித செயற்குறிகள் என்பது இரு செயலேற்பிகளை ஏற்றுக் கொண்டு அதன் மீது கணித செயல்பாடுகளை செய்யப் பயன்படுகிறது.

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=100, b=10 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
+ (கூட்டல்)	>>> a + b	110
- (கழித்தல்)	>>> a - b	90
* (பெருக்கல்)	>>> a * b	1000
/ (வகுத்தல்)	>>> a / b	10.0
% (வகுமீதி)	>>> a % b	10
** (அடுக்கு)	>>> a ** b	10000
// (முழு எண் தொகுதி)	>>> a // b	3

2. பைத்தானில் மதிப்பிடுத்து செயற்குறிகள் என்றால் என்ன?

பைத்தானில் = என்பது ஒரு மதிப்பிடுத்து செயற்குறியாகும். இது மாறிகளுக்கு மதிப்பிடுத்த பயன்படுகிறது. பைத்தான் +=, -=, \*=, /=, %=, \*\*= மற்றும் //= போன்ற கூட்டுச் செயற்குறிகளை ஏற்கும்.

Operator	Example	Result
x = 10 எனில்		
=	>>> x = 10 >>> b = "Computer"	10 Computer
+=	>>>x+=20 #x=x+20	30
-=	>>>x-=5 #x=x-5	25
*=	>>>x*=5 #x=x*5	125

/=	>>>x/=2	#x=x/2	62.5
%=	>>>x%=3	#x=x%3	2.5
**=	>>>x**=2	#x=x**2	6.25
//=	>>>x//=3	#x=x//3	2.0

### 3. மும்ம செயற்குறியை எடுத்துக்காட்டுடன் எழுதுக.

மும்ம செயற்குறி நிபந்தனை செயற்குறி என்றும் அழைக்கப்படுகிறது. இது சமன்பாடுகளின் நிபந்தனையை சரியா அல்லது தவறா என்று சோதித்து செயல்படுத்தும். மும்ம செயற்குறி பலவரி if..else கூற்று போல் அல்லாது நிபந்தனைகளை ஒற்றை வரியில் சோதிக்க அனுமதிக்கிறது.

மும்ம செயற்குறிக்கான தொடரியல்:

மாறியின் பெயர் = [on\_true] if [பரிசோதிப்பு கோவை] else [on\_false]

#### உதாரணம்:

min = 50 if 49<50 else 70 // min = 50  
min = 50 if 49>50 else 70 // min = 70

### 4. விடுபடு வரிசைப் பற்றி எழுதி எடுத்துக்காட்டுடன் தருக.

பைத்தானில் பின் சாய்வுகோடு ஆகிய சிறப்பு குறியீட்டை "விடுபடு" குறியுரு என்று அழைக்கிறோம். இது சில வெற்று இடைவெளியை குறிப்பிட பயன்படுகிறது.

விடுபடு தொடர் வரிசை	விளக்கம்	எடுத்துக்காட்டு	வெளியீடு
\\	பின்சாய்வுக்கோடு	>>> print("\\test")	\test
\'	ஒற்றை மேற்கோள்	>>> print("Doesn't")	Doesn't
\"	இரட்டை மேற்கோள்	>>> print("\"Python\"")	"Python"
\n	புதிய வரி	print("Python","\n", "Lang..")	Python Lang..
\t	தத்தல்	print("Python","\t", "Lang..")	Python   Lang..

### 5 மதிப்பெண் வினாக்கள்: (புத்தக வினாக்கள்)

#### 1. செயல்முறை ஸ்கிரிப்ட் முறைமை நிரலாக்கம் பற்றி எழுதுக.

ஸ்கிரிப்ட் என்பது பைத்தான் கட்டளைகளை கொண்ட ஒரு உரை ஆவணத்தைக் குறிக்கிறது.

#### பயன்கள்:

- குறிமுறை பயனாக்கம்
- ஒருமுறை எழுதிய ஸ்கிரிப்டை பலமுறை பயன்படுத்திக் கொள்ளலாம்.
- பதிப்பாய்வு செய்ய முடியும்.

#### ஸ்கிரிப்ட் எழுதுதல்

- பைத்தான் சாளரத்தில் File → New File (அ) Ctrl + N
- வெற்று ஸ்கிரிப்ட் எடிட்டர் தோன்றும்.
- கட்டளைகளை உள்ளிடவும்.

#### உதாரணம்:

a = 100  
b = 200

```
c = a+b
print("The sum=", c)
```

### ஸ்கிரிப்ட்டை சேமித்தல்:

- File → Save (அ) Ctrl + S
- Save as உரையாடல் பெட்டி தோன்றும்.
- கோப்பு சேமிக்கப்பட வேண்டிய இடத்தை தேர்ந்தெடுக்க வேண்டும்.
- File Name பெட்டியில் கோப்பின் பெயரை உள்ளிட வேண்டும். கோப்புகள் முன்னியல்பாக **.py** என்ற நீட்டிப்புடன் சேமிக்கப்படுகிறது.
- Save பொத்தானை அழுத்தவும்.

### ஸ்கிரிப்ட்டை இயக்குதல்:

- Run → Run Module (அ) F5 யை தேர்ந்தெடுக்கவும்.
- பிழைகள் இருப்பின், IDLE திரையில் காண்பிக்கப்படும்.
- பிழைகளை திருத்துவதற்கு ஸ்கிரிப்ட் எடிட்டர்க்கு சென்று, பிழைகளை திருத்தவும். கோப்பை சேமித்து மீண்டும் இயக்கவும்.
- பிழைகள் இல்லாத பைத்தான் குறிமுறையை இயக்கியவுடன் அதன் வெளியீடு IDLE திரையில் தோன்றும்.

## 2. input() மற்றும் print() செயற்கூறுகள் பற்றி விளக்குக.

### input() செயற்கூறு:

பைத்தானில் நிரலை இயக்கும் போது தரவுகளை உள்ளீடாகப் பெற்று கொள்ள பயன்படும் செயற்கூறு input() செயற்கூறு ஆகும்.

### பொதுவடிவம்:

```
மாறி = input ("உள்ளிடப்பட வேண்டிய சரம்")
```

- prompt string என்பது பயனர் கொடுக்கப்பட வேண்டிய உள்ளீடு எது என்பதை உணர்த்தும் கூற்று ஆகும். இது கொடுக்கப்படவில்லையென்றால் திரையில் எந்த தகவலும் தோன்றாது.
- input() செயற்கூறு விசைப்பலகையின் மூலம் தட்டச்சு செய்தவற்றை மாறியில் சேமித்து வைக்கும்.

### எடுத்துக்காட்டு:

```
>>>city = input("Enter your city")
Enter your city: TamilNadu
>>>print("I am from", city)
I am from TamilNadu
```

- input() செயற்கூறு அனைத்து உள்ளீடுகளையும் சரங்கள் (அ) குறியீடுவாக ஏற்றுக் கொள்ளும். ஆனால் எண்களாக அல்ல.
  - int() செயற்கூறு அனைத்து சரங்களை எண் வகையாக மாற்ற பயன்படுகிறது.
- எடுத்துக்காட்டு:      x = int ( input ("Enter number:" ) )

### print () செயற்கூறு:

- பைத்தானில் நிரலை இயக்கும் போது தரவுகளை வெளியிட print() செயற்கூறு பயன்படுகிறது.
- print() செயற்கூறு தீர்வுகளை திரையில் காண்பிக்கும் முன் கோவையை மதிப்பீடு செய்யும்.
- காற்புள்ளி (,) கொண்டு print () செயற்கூறின் ஒன்றுக்கு மேற்பட்ட உறுப்புகளைப் பிரிக்கலாம்.

**பொதுவடிவம்:**

```
print("string")
print(variable)
print("string",variable)
print("string1",variable1, "string2", variable2)
```

**எடுத்துக்காட்டு:**

```
>>> print("Welcome")
Welcome
```

```
>>> x=5
>>> y=10
>>> z = x + y
>>> print(z)
15
```

```
>>> print("Sum= ",z)
Sum = 15
```

**3. பைத்தானில் உள்ள வில்லைகள் பற்றி எழுதுக.**

பைத்தான் நிரலில் இடம்பெறும் வரிகளை அடிப்படை சொற்களாக பிரிக்கிறது. இந்த கூறுகள் வில்லைகள் எனப்படும். பொதுவாக வில்லைகளின் வகைகளாவன:

- 1) குறிப்பெயர்கள்
- 2) சிறப்புச்சொற்கள்
- 3) செயற்குறிகள்
- 4) வரம்புக்குறிகள்
- 5) நிலையுருக்கள்

**1) குறிப்பெயர்கள்:**

மாறி, செயற்கூறு, இனக்குழு தொகுதி அல்லது பொருளின் பெயர்களை குறிப்பெயர் என்று அழைக்கப்படுகிறது.

- ◆ குறிப்பெயர்கள் எழுத்துக்கள் (A..Z or a..z) அல்லது அடிக்கீறல் (\_) கொண்டு தொடங்க வேண்டும்.
- ◆ குறிப்பெயர்கள் எண்கள் (0..9) கொண்டிருக்கலாம்.
- ◆ பைத்தான் குறிப்பெயர்கள் எழுத்து வடிவுணர்வு கொண்டது.
- ◆ குறிப்பெயர்கள் ஆனது பைத்தான் சிறப்புச்சொற்களாக இருக்கக்கூடாது.
- ◆ பைத்தானில் %, \$, @ etc போன்ற நிறுத்தற்குறிகள் குறிப்பெயர்களில் இடம்பெறக் கூடாது.

**எடுத்துக்காட்டுகள்:**

```
sum, total_marks, regno, num1
```

**2) சிறப்புச்சொற்கள்:**

நிரலின் அமைப்பை அடையாளம் காண பைத்தான் மொழிப்பெயர்ப்பி சிறப்புச்சொற்களை பயன்படுத்துகிறது. சிறப்புச் சொற்கள் மொழிப்பெயர்ப்பியில் குறிப்பிட்ட பொருள் கொண்டுள்ளதால் இவற்றை பிற பயன்பாட்டிற்கு பயன்படுத்தக் கூடாது.



6. \_\_\_ முறைமை பைத்தான் தூண்டுக்குறியில் (>>>) நிரலை எழுத உதவும்.  
அ)ஸ்கிரிப்ட் ஆ)ஊடாகும் இ)கட்டளை ஈ)அ மற்றும் ஆ
7. \_\_\_ முறைமை ஒரு எளிய முறை கால்குலேட்டரை போல் பயன்படுத்தலாம்  
அ)ஸ்கிரிப்ட் ஆ)ஊடாகும் இ)கட்டளை ஈ)அ மற்றும் ஆ
8. \_\_\_ முறைமையில் நிரல் குறிமுறை தனி கோப்பாக, **.py** என்ற நீட்டிப்புடன் சேமிக்கப்பட்டு இயக்கப்படும்  
அ)ஸ்கிரிப்ட் ஆ)ஊடாகும் இ)கட்டளை ஈ)அ மற்றும் ஆ
9. \_\_\_ முறைமை பைத்தான் மூல நிரலை உருவாக்கவும், பதிப்பாய்வு செய்யவும் பயன்படுகிறது.  
அ)ஸ்கிரிப்ட் ஆ)ஊடாகும் இ)கட்டளை ஈ)அ மற்றும் ஆ
10. பின்வருபவைகளில் எந்த கட்டளைபைத்தான் IDLE யை துவக்கப் பயன்படுகிறது?  
அ) Start → All Programs → IDLE (Python 3.x)  
ஆ) Start → All Programs → IDLE (Python 3.x) → Python 3.x  
இ) Start → All Programs → Python 3.x → (Python 3.x)  
ஈ) Start → All Programs → Python 3.x → IDLE (Python 3.x)
11. பைத்தானில் புதிய ஸ்கிரிப்ட்டினை எவ்வாறு உருவாக்குவாய்?  
அ)File → Save ஆ)File → New இ)New → File ஈ)Save → File
12. பைத்தானில் ஸ்கிரிப்ட்டினை எவ்வாறு சேமிப்பாய்?  
அ)File → Save ஆ)Ctrl + S இ)New → File ஈ)அ அல்லது ஆ
13. கொடாநிலையாக பைத்தான் ஸ்கிரிப்ட் \_\_\_ நீட்டிப்புடன் சேமிக்கப்படுகிறது.  
அ).python ஆ).pyt இ).py ஈ)மேற்கண்ட எதுமில்லை
14. பைத்தானில் ஸ்கிரிப்ட்டினை எவ்வாறு இயக்குவாய்?  
அ)Run → Run Module ஆ) Run Module இ)F5 ஈ)அ அல்லது இ
15. பைத்தானில் எந்த செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடாக பெற்றுக் கொள்ளப் பயன்படுகிறது?  
அ)int() ஆ)input() இ) print() ஈ) function()
16. பைத்தானில் எந்த செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை வெளியிட பயன்படுகிறது?  
அ)int() ஆ)input() இ) print() ஈ) function()
17. \_\_\_ கொண்டு print() செயற்கூறினுள் ஒன்றுக்கு மேற்பட்ட உறுப்புகளைப் பிரிக்கலாம்.  
அ).(புள்ளி) ஆ)(:):முக்காற்புள்ளி இ)(:):அரைப்புள்ளி ஈ)(,) காற்புள்ளி
18. பைத்தானில் எந்த செயற்கூறு சரவகை தரவை எண்வகை தரவாக வெளிப்படையாக மாற்ற உதவுகிறது?  
அ)int() ஆ)input() இ) print() ஈ) function()
19. பைத்தான் நிரலில் இடம்பெறும் வரிகளை அடிப்படை சொற்களாக பிரிக்கிறது. இந்தக் கூறுகள் \_\_\_ எனப்படும்  
அ)செயற்கூறிகள் ஆ)சிறப்புச்சொற்கள் இ)வரம்புக்குறிகள் ஈ)வில்லைகள்
20. மாறி, செயற்கூறு, இனக்குழு, தொகுதி அல்லது பொருளின் பெயர்கள் \_\_\_ என்று அழைக்கப்படுகிறது.  
அ)செயற்கூறிகள் ஆ)சிறப்புச்சொற்கள் இ)வரம்புக்குறிகள் ஈ)குறிப்பெயர்கள்
21. பின்வருபவைகளில் எது சரியான குறிப்பெயராகும்?  
அ)12name ஆ) name\$ இ) total\_marks ஈ) continue
22. பின்வருபவைகளில் எது தவறான குறிப்பெயராகும்?  
அ)name ஆ) regno இ) total-marks ஈ) num1

23. நிரலின் அமைப்பை அடையாளம் காண பைத்தான் மொழிப்பெயர்ப்பி \_\_\_ களை பயன்படுத்துகிறது  
 அ)செயற்குறிகள் ஆ)சிறப்புச்சொற்கள் இ)வரம்புக்குறிகள் ஈ)குறிப்பெயர்கள்
24. கணிப்பொறி நிரலாக்க மொழியில் \_\_\_ என்ற சிறப்பு குறியீடுகள் கணிப்பீடுகள், நிபந்தனை சோதிப்பு போன்ற செயல்பாடுகளை செய்யப் பயன்படுகிறது  
அ)செயற்குறிகள் ஆ)சிறப்புச்சொற்கள் இ)வரம்புக்குறிகள் ஈ)குறிப்பெயர்கள்
25. எந்த செயற்குறிகள், இரு செயலேற்பிகளை ஏற்றுக்கொண்டு அதன் மீது கணித செயல்பாடுகளை செய்யும்?  
 அ)தருக்க ஆ)ஒப்பீட்டு இ)கணித ஈ)நிபந்தனை
26.  $a=100$  எனில்  $a**2$  ன் மதிப்பு யாது?  
 அ)100 ஆ)1000 இ)10 ஈ)10000
27.  $a=100$  எனில்  $a//30$  ன் மதிப்பு யாது?  
 அ)3.3 ஆ)0.3 இ)30 ஈ)3
28.  $a=100$ ,  $b=10$  எனில்  $a\%b$  ன் மதிப்பு யாது?  
 அ)100 ஆ)0 இ)10 ஈ)10000
29.  $a=100$ ,  $b=10$  எனில்  $a/b$  ன் மதிப்பு யாது?  
 அ)100 ஆ)10.0 இ)0 ஈ)10000
30. எந்த செயற்குறிகள் ஒப்பீடு செயற்குறிகள் என்று அழைக்கப்பட்டு, இரு செயலேற்பிகளுக்கு இடையேயான உறவுமுறையை சோதித்தறிய உதவுகிறது?  
 அ)தருக்க ஆ)தொடர்புடைய இ)கணித ஈ)நிபந்தனை
31.  $a=100$ ,  $b=35$  எனில்  $a==b$  ன் மதிப்பு யாது?  
அ)தவறு ஆ)சரி இ)அ மற்றும் ஆ ஈ)ஏதுமில்லை
32.  $a=100$ ,  $b=35$  எனில்  $a!=b$  ன் மதிப்பு யாது?  
 அ)தவறு ஆ)சரி இ)அ மற்றும் ஆ ஈ)ஏதுமில்லை
33. பைத்தானில் எந்த செயற்குறி கொடுக்கப்பட்ட ஒப்பீடு கோவையின் மீது தருக்க செயல்பாடுகளை மேற்கொள்ள பயன்படுகிறது?  
அ)தருக்க ஆ)தொடர்புடைய இ)கணித ஈ)நிபந்தனை
34.  $a=97$ ,  $b=35$  எனில்  $a>b$  and  $a==b$  ன் மதிப்பு யாது?  
அ)தவறு ஆ)சரி இ)அ மற்றும் ஆ ஈ)ஏதுமில்லை
35.  $a=97$ ,  $b=35$  எனில்  $a>b$  or  $a==b$  ன் மதிப்பு யாது?  
 அ)தவறு ஆ)சரி இ)அ மற்றும் ஆ ஈ)ஏதுமில்லை
36.  $a=97$ ,  $b=35$  எனில் not  $a>b$  ன் மதிப்பு யாது?  
அ)தவறு ஆ)சரி இ)அ மற்றும் ஆ ஈ)ஏதுமில்லை
37. பின்வரும் எந்த செயற்குறி மாறிகளுக்கு மதிப்பிருத்த பயன்படுகிறது?  
 அ)தருக்க ஆ)தொடர்புடைய இ)மதிப்பிருத்து ஈ)நிபந்தனை
38. பின்வருபவைகளில் எது மதிப்பிருத்து செயற்குறியாகும்?  
 அ)== ஆ) != இ) = ஈ) !==
39. பின்வருபவைகளில் எது சமமில்லை என்பதற்கான செயற்குறியாகும்?  
அ) != ஆ) == இ) = ஈ) !==
40.  $a=10$  எனில்  $a/=3$  ன் மதிப்பு யாது?  
 அ)1 ஆ)3.3 இ)3 ஈ)10
41.  $a=10$  எனில்  $a\%=3$  ன் மதிப்பு யாது?  
அ)1 ஆ)3.3 இ)3 ஈ)10



**பைத்தான் ஸ்கிரிப்ட்டை சேமித்தல்:**

- File → Save (அ) Ctrl + S என்று கிளிக் செய்தால் Save as உரையாடல் பெட்டி தோன்றும்.
- கோப்பு சேமிக்கப்பட வேண்டிய இடத்தை தேர்ந்தெடுத்து, File Name பெட்டியில் கோப்பின் பெயரை உள்ளிட வேண்டும். கோப்புகள் முன்னியல்பாக **.py** என்ற நீட்டிப்புடன் சேமிக்கப்படுகிறது. Save பொத்தானை அழுத்தவும்.

**4. பைத்தான் ஸ்கிரிப்ட்டை எவ்வாறு இயக்குவாய்?**

**பைத்தான் ஸ்கிரிப்ட்டை இயக்குதல்:**

- Run → Run Module (அ) F5 யை தேர்ந்தெடுக்கவும்.
- பிழைகள் இருப்பின், IDLE திரையில் காண்பிக்கப்படும்.
- பிழைகளை திருத்துவதற்கு ஸ்கிரிப்ட்டை எடிட்டர்க்கு சென்று, பிழைகளை திருத்தவும். கோப்பை சேமித்து மீண்டும் இயக்கவும்.
- பிழைகள் இல்லாத பைத்தான் குறிமுறையை இயக்கியவுடன் அதன் வெளியீடு IDLE திரையில் தோன்றும்.

**5. பைத்தானில் குறிப்புரை பற்றி எழுதுக.**

பைத்தானில் குறிப்புரை ஹாஷ் (#) என்ற குறியுடன் தொடங்கும். # குறியுடன் தொடங்கும் வரிகள் குறிப்புரையாக கருதப்பட்டு, பைத்தான் மொழிப்பெயர்ப்பி இவற்றை குறிமுறையின் பகுதியாக கருதாது. குறிப்புரைகள் ஒரு வரி அல்லது பல வரியாக இருக்கலாம்.

**உதாரணம்:**

```
# it is single line comment
# it is multi line comment
Contains more than one line #
```

**6. குறிப்பெயர்கள் என்றால் என்ன?**

மாறி, செயற்கூறு, இனக்குழு, தொகுதி அல்லது பொருளின் பெயர்களை குறிப்பெயர் என்று அழைக்கப்படுகிறது

**7. சிறப்புச்சொற்கள் என்றால் என்ன?**

நிரலின் அமைப்பை அடையாளம் காண பைத்தான் மொழிப்பெயர்ப்பி பயன்படுத்துவது சிறப்புச்சொற்கள் எனப்படும். உதாரணம்: is, def, as, or etc....

**8. பைத்தானில் உள்ள ஒப்பீடு செயற்குறிகளை எழுதுக.**

பைத்தானில் தொடர்புடைய செயற்குறிகள் ஒப்பீடு செயற்குறிகள் என்றும் அழைக்கப்படுகின்றன. இது ஒரு செயலேற்பிகளுக்கு இடையேயான உறவுமுறையை சோதித்து அறிய உதவும். ஒப்பீடு செயற்குறிகள் பின்ருமாறு:

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=100, b=35 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
== (நிகர்)	>>> a == b	தவறு
> (விடப்பெரியது)	>>> a > b	சரி

>= (விடப்பெரியது அல்லது நிகர்)	>>> a >= b	சரி
< (விடச்சிறியது)	>>> a < b	தவறு
<= (விடச்சிறியது அல்லது நிகர்)	>>> a <= b	தவறு
!= (நிகரில்லை)	>>> a != b	சரி

### 9. பைத்தானில் வரம்புக்குறிகள் என்றால் என்ன?

பைத்தான் ஆனது குறியீடு அல்லது குறியீடுகளின் தொகுப்பை, கோவை, பட்டியல், அகராதி மற்றும் சரங்களில் பயன்படுத்துகிறது. பின்வருவன பைத்தான் வரம்புக்குறிகளாகும்.

(	)	[	]	{	}
,	:	.	'	=	;
+=	--	*=	/=	//=	%=
&=	=	^=	>>=	<<=	**=

### 3 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

#### 1. பைத்தானின் சிறப்பம்சங்கள் யாவை?

பைத்தான் சிறப்பம்சங்கள்:

- பைத்தான் ஒரு பொது பயன்பாட்டு மொழியாகும். இதை அறிவியல் அல்லாத நிரலாக்கத்திற்கு பயன்படுத்தலாம்.
- இது இயக்க முறையை சாராத நிரலாக்க மொழியாகும்.
- பைத்தான் நிரல் எளிதாக புரிந்து கொள்ள இயலும்.

#### 2. input() மற்றும் print() செயற்கூறுகள் பற்றி குறிப்பு எழுதுக.

**input() செயற்கூறு:**

பைத்தானில் நிரலை இயக்கும் போது தரவுகளை உள்ளீடாகப் பெற்று கொள்ள பயன்படும் செயற்கூறு input() செயற்கூறு ஆகும்.

பொதுவடிவம்:

```
மாறி = input("உள்ளிடப்பட வேண்டிய உரை")
```

எடுத்துக்காட்டு:

```
>>>city = input("Enter your city")
Enter your city: TamilNadu
```

**print () செயற்கூறு:**

- பைத்தானில் நிரலை இயக்கும் போது தரவுகளை வெளியிட print() செயற்கூறு பயன்படுகிறது.

பொதுவடிவம்:

```
print("string")
print(variable)
print("string",variable)
print("string1",variable1, "string2", variable2)
```

எடுத்துக்காட்டு:

```
>>> print("Welcome")
Welcome
```

3. பைத்தானில் உள்ள தொடர்புடைய அல்லது ஒப்பீடு செயற்குறிகளைப் பற்றி எழுதுக.

பைத்தானில் தொடர்புடைய செயற்குறிகள் ஒப்பீடு செயற்குறிகள் என்றும் அழைக்கப்படுகின்றன. இது ஒரு செயலேற்பிகளுக்கு இடையேயான உறவுமுறையை சோதித்து அறிய உதவும். ஒப்பீடு செயற்குறிகள் பின்ருமாறு:

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=100, b=35 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
== (நிகர்)	>>> a == b	தவறு
> (விடப்பெரியது)	>>> a > b	சரி
>= (விடப்பெரியது அல்லது நிகர்)	>>> a >= b	சரி
< (விடச்சிறியது)	>>> a < b	தவறு
<= (விடச்சிறியது அல்லது நிகர்)	>>> a <= b	தவறு
!= (நிகரில்லை)	>>> a != b	சரி

4. பைத்தானில் உள்ள தருக்க செயற்குறிகளைப் பற்றி எழுதுக.

பைத்தானில் தருக்க செயற்குறிகள் கொடுக்கப்பட்ட ஒப்பீடு கோவையின் மீது தருக்க செயல்பாடுகளை மேற்கொள்ள பயன்படுகிறது. and, or மற்றும் not ஆகிய மூன்று தருக்க செயற்குறிகள் உள்ளன.

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=97, b=35 என மதிப்பாக கொண்டு பின்வரும் தருக்க கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
or	>>> a > b or a==b	சரி
and	>>> a > b and a==b	தவறு
not	>>> not a > b	தவறு

5. பைத்தானில் உள்ள மதிப்பிடுத்து செயற்குறிகளை எழுதுக.

பைத்தானில் = என்பது மதிப்பிடுத்து செயற்குறியாகும். இது மாறிகளுக்கு மதிப்பிடுத்த பயன்படுகிறது. மதிப்பிடுத்து செயற்குறிகள் பின்ருமாறு:

செயற்குறி	விளக்கம்	எடுத்துக்காட்டு	விடை
x=10 எனில்			
=	வலப்பக்கமுள்ள செயலேற்பினை இடப்பக்கமுள்ள மாறிக்கு இருத்தும்.	>>>x=10	10
+=	செயலேற்பிகளை கூட்டி வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x+=20 #x=x+20	30

=	செயலேற்பிகளை கழித்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x-=5 #x=x-5	25
*=	செயலேற்பிகளை பெருக்கி வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x*=5 #x=x*5	125
/=	செயலேற்பிகளை வகுத்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x/=2 #x=x/2	62.5
%=	இரு செயலேற்பிகள் மீது செயற்பட்டு, வகுமீதியை கணக்கிட்டு விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x%=3 #x=x%3	2.5
**=	செயலேற்பியில் அடுக்குறியை செயற்படுத்தி இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்	>>>x**=2 #x=x**2	6.25
//=	செயலேற்பியின் மீது முழு எண் வகுத்தல் மதிப்பிட்டு இடப்பக்க செயலேற்பியில் இருத்தும்.	>>>x//=3 #x=x//3	2.0

#### 5 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

#### 1. பைத்தானில் உள்ள பல்வேறு வகையான செயற்குறிகளை பட்டியலிடுக.

பைத்தானில் செயற்குறிகள் என்பவை குறியீடுகள். இவை செயலேற்பியில் உள்ள மதிப்புகளில் செயல்படும். கணிப்பொறி நிரலாக்க மொழியில் செயற்குறிகள் ஆனது என்ற சிறப்பு குறியீடுகள் கணிப்பீடுகள், நிபந்தனை சோதிப்பு போன்ற செயல்பாடுகளை செய்யப் பயன்படுகிறது. செயற்குறிகளின் வகைகள் பின்வருமாறு:

- 1) கணித செயற்குறிகள்
- 2) தொடர்புடைய அல்லது ஒப்பீடு செயற்குறிகள்
- 3) தடுக்க செயற்குறிகள்
- 4) மதிப்பிடுத்து செயற்குறிகள்
- 5) நிபந்தனை செயற்குறி

#### கணித செயற்குறிகள்

கணித செயற்குறிகள் என்பது இரு செயலேற்பிகளை ஏற்றுக் கொண்டு அதன் மீது கணித செயல்பாடுகளை செய்யப் பயன்படுகிறது.

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=100, b=10 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
+ (கூட்டல்)	>>> a + b	110
- (கழித்தல்)	>>> a - b	90

* (பெருக்கல்)	>>> a * b	1000
/ (வகுத்தல்)	>>> a / b	10.0
% (வகுமீதி)	>>> a % b	10
** (அடுக்கு)	>>> a ** b	10000
// (முழு எண் தொகுதி)	>>> a // b	3

### ஒப்பீட்டு செயற்குறிகள்:

பைத்தானில் தொடர்புடைய செயற்குறிகள் ஒப்பீடு செயற்குறிகள் என்றும் அழைக்கப்படுகின்றன. இது ஒரு செயலேற்பிகளுக்கு கிடையேயான உறவுமுறையை சோதித்து அறிய உதவும். ஒப்பீடு செயற்குறிகள் பின்ருமாறு:

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=100, b=35 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
== (நிகர்)	>>> a == b	தவறு
> (விடப்பெரியது)	>>> a > b	சரி
>= (விடப்பெரியது அல்லது நிகர்)	>>> a >= b	சரி
< (விடச்சிறியது)	>>> a < b	தவறு
<= (விடச்சிறியது அல்லது நிகர்)	>>> a <= b	தவறு
!= (நிகரில்லை)	>>> a != b	சரி

### தருக்க செயற்குறிகள்:

பைத்தானில் தருக்க செயற்குறிகள் கொடுக்கப்பட்ட ஒப்பீடு கோவையின் மீது தருக்க செயல்பாடுகளை மேற்கொள்ள பயன்படுகிறது. and, or மற்றும் not ஆகிய மூன்று தருக்க செயற்குறிகள் உள்ளன.

செயற்குறி	எடுத்துக்காட்டு	தீர்வு
a=97, b=35 என மதிப்பாக கொண்டு பின்வரும் தருக்க கோவையை மதிப்பிட கிடைக்கும் தீர்வுகள் பின்வருமாறு		
or	>>> a > b or a == b	சரி
and	>>> a > b and a == b	தவறு
not	>>> not a > b	தவறு

### மதிப்பிடுத்து செயற்குறிகள்:

பைத்தானில் = என்பது மதிப்பிடுத்து செயற்குறியாகும். இது மாறிகளுக்கு மதிப்பிடுத்த பயன்படுகிறது. மதிப்பிடுத்து செயற்குறிகள் பின்ருமாறு:

செயற்குறி	விளக்கம்	எடுத்துக்காட்டு	விடை
x=10 எனில்			
=	வலப்பக்கமுள்ள செயலேற்பினை இடப்பக்கமுள்ள மாறிக்கு இருத்தும்.	>>>x=10	10
+=	செயலேற்பிகளை கூட்டி வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x+=20 #x=x+20	30
-=	செயலேற்பிகளை கழித்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x-=5 #x=x-5	25
*=	செயலேற்பிகளை பெருக்கி வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x*=5 #x=x*5	125
/=	செயலேற்பிகளை வகுத்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x/=2 #x=x/2	62.5
%=	இரு செயலேற்பிகள் மீது செயற்பட்டு, வகுமீதியை கணக்கிட்டு விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x%=3 #x=x%3	2.5
**=	செயலேற்பியில் அடுக்குறியை செயற்படுத்தி இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும்.	>>>x**=2 #x=x**2	6.25
//=	செயலேற்பியின் மீது முழு எண் வகுத்தல் மதிப்பிட்டு இடப்பக்க செயலேற்பியில் இருத்தும்.	>>>x//=3 #x=x//3	2.0

### நிபந்தனை செயற்குறி:

மும்ம செயற்குறி நிபந்தனை செயற்குறி என்றும் அழைக்கப்படுகிறது. இது சமன்பாடுகளின் நிபந்தனையை சரியா அல்லது தவறா என்று சோதித்து செயல்படுத்தும். மும்ம செயற்குறி பலவரி if..else கூற்று போல் அல்லாது நிபந்தனைகளை ஒற்றை வரியில் சோதிக்க அனுமதிக்கிறது.

### மும்ம செயற்குறிக்கான தொடரியல்:

மாறியின் பெயர் = [on\_true] if [பரிசோதிப்பு கோவை] else [on\_false]

### உதாரணம்:

min = 50 if 49<50 else 70 // min = 50

min = 50 if 49>50 else 70 // min = 70



2. **break** கூற்றைப் பற்றி குறிப்பு வரைக.

break கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது. நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது.

3. **if..else** கூற்றின் பொதுவடிவத்தை எழுதுக.

**if..else** கூற்றின் பொதுவடிவம்:

```
if <நிபந்தனை>:
    கூற்று_தொகுதி1
else:
    கூற்று_தொகுதி2
```

4. கட்டுப்பாட்டு கட்டமைப்பு என்றால் என்ன?

கட்டுப்பாட்டு நிரலின் ஒரு பகுதியிலிருந்து மற்றொரு பகுதிக்கு தாவுவதற்கு காரணமாக நிரல் கூற்றுகள் கட்டுப்பாட்டு கட்டமைப்பு அல்லது கட்டுப்பாட்டு கூற்றுகள் எனப்படும்.

5. **range()** செயற்கூறு பற்றி குறிப்பு வரைக. **\*\*\* (March 2020)**

பைத்தானில் for மடக்கில் வரிசையில் உள்ள தொடக்க, இறுதி, மதிப்புகளைக் குறிப்பதற்காக பயன்படும் செயற்கூறு range() ஆகும். range() செயற்கூறு start முதல் stop -1 வரையிலான மதிப்பு பட்டியல்களை உருவாக்குகிறது.

**range()** தொடரியல்:

*range (start, stop, [step])*

3 மதிப்பெண் வினாக்கள்:

1. பின்வரும் வெளியீட்டைப் பெற நிரலை எழுதுக.

A  
A B  
A B C  
A B C D  
A B C D E

**Program:**

```
for i in range(1,6):
    x=65
    for j in range(1,i+1):
        print(chr(x), end=" ")
        x+=1
    print(end='\n')
```

2. **if..else** அமைப்பைப் பற்றி குறிப்பு வரைக.

if..else கூற்றானது சரி தொகுதி மற்றும் தவறு தொகுதி இரண்டையுமே சரிபார்ப்பதற்கான கட்டுப்பாட்டை வழங்குகிறது.

பொதுவடிவம்:

```
if <நிபந்தனை>:
    கூற்று_தொகுதி1
else:
    கூற்று_தொகுதி2
```

உதாரணம்:

```
a=int(input("Enter a number:"))
if a%2==0:
    print("The given number is Even")
else:
    print("The given number is Odd")
```

வெளியீடு:

```
Enter a number:7
The given number is Odd
```

if..else இரு மாற்று வழிகளை வழங்குகிறது. நிபந்தனையானது எந்த தொகுதியை நிறைவேற்றப்பட வேண்டும் என்பதை தீர்மானிக்கிறது.

3. if..else..elif கூற்றைப் பயன்படுத்தி கொடுக்கப்பட்ட மூன்று எண்களில் பெரிய எண்ணைக் கண்டுபிடிப்பதற்கான பொருத்தமான நிரலை எழுதுக.

Program:

```
a=int(input("Enter a value:"))
b=int(input("Enter b value:"))
c=int(input("Enter c value:"))
if a>b and a>c:
    print("A is Big")
elif b>a and b>c:
    print("B is Big")
else:
    print("C is Big")
```

Output:

```
Enter a value:35
Enter b value:43
Enter c value:12
B is Big
```

4. while மடக்கின் பொதுவடிவம் யாது?

பொதுவடிவம்:

```
while <நிபந்தனை>:
    கூற்று_தொகுதி1
[else:                #optional
    கூற்று_தொகுதி2]
```

Example:

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

i=i+1

**Output:**

1 2 3 4 5

**5. break மற்றும் continue கூற்றுகளின் வேறுபாடு யாது?**

<i>break</i>	<i>continue</i>
break கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது.	continue கூற்றானது மடக்கின் மீதமுள்ள குறிமுறையைத் தவிர்த்து அடுத்த மடக்கு செயலை ஆரம்பிக்கும்
பொதுவுடிவம்: <i>break</i>	பொதுவுடிவம்: <i>continue</i>

**5 மதிப்பெண் வினாக்கள்:**

**1. for மடக்கைப் பற்றி விரிவான விடையளிக்கவும்.**

- for மடக்கு பயன்படுத்தக்கூடிய ஓர் மடக்காகும்.
- இது ஒரு நுழைவு சோதிப்பு மடக்காகும்.
- நிபந்தனை முதலிலேயே சோதிக்கப்பட்டு சரி எனில் மடக்கின் உடற்பகுதி செயல்பாட்டுத் தொகுதியை நிறைவேற்றும். இல்லையெனில் மடக்கு நிறைவேற்றாமல் வெளியேற்றப்படும்.

**for மடக்கின் பொதுவுடிவம்:**

**for** தொடரியல்\_மாறி **in** வரிசை:  
கூற்று\_தொகுதி 1  
  
[else: #optional block  
கூற்று\_தொகுதி 2 ]

- தொடரியலில் குறிப்பிட்டுள்ள தொடரியல்\_மாறியானது (counter\_variable), மடக்கில் பயன்படுத்தப்படும் கட்டுப்பாட்டு மாறியை ஒத்ததாகும்.
- வரிசை என்பது தொடக்க, இறுதி மற்றும் மிகுப்பு மதிப்புகளைக் குறிக்கும்.
- பைத்தானில் for மடக்கில் வரிசையில் உள்ள தொடக்க, இறுதி மதிப்புகளைக் குறிப்பதற்கு range() செயற்கூறு பயன்படுகிறது.
- range() செயற்கூறு start முதல் stop-1 வரையிலான மதிப்பு பட்டியலை உருவாக்குகிறது.

**range() ன் தொடரியல்:**

*range(start, stop, [step])*

இதில்,

*start* - என்பது தொடக்க மதிப்பைக் குறிக்கும்

*stop* - என்பது இறுதி மதிப்பைக் குறிக்கும்

*step* - மிகுப்பு மதிப்பைக் குறிக்கும். இது விருப்பப் பகுதியாகும்.

**உதாரணம்:**

```
for i in range(2,10,2):
    print(i,end=' ')
else:
    print("\nEnd of loop")
```

வெளியீடு:

2 4 6 8  
End of loop

**2. if..else..elif கூற்றை எடுத்துக்காட்டுடன் விளக்குக.**

if கூற்றுகளைத் தொடர் கூற்றுகளாக அமைக்கும் விரும்பும் போது 'else' பகுதிக்குப் பதிலாக 'elif' பகுதி பயன்படுத்தலாம்.

தொடரியல்:

```
if <நிபந்தனை_1>:
    கூற்று_தொகுதி 1
elif<நிபந்தனை_2>:
    கூற்று_தொகுதி 2
else:
    கூற்று_தொகுதி n
```

- மேற்கண்ட if..elif..else பொதுவடிவத்தில், நிபந்தனை\_1 பரிசோதிக்கப்பட்டு சரி எனில் கூற்று\_தொகுதி 1 செயல்பட்டு நிபந்தனை\_1 நிறைவேற்றப்படும்.
- இல்லையெனில் கட்டுப்பாடு, நிபந்தனை\_2 ஐ பரிசோதித்து, சரி எனில் கூற்று\_தொகுதி 2 நிறைவேற்றும்.
- இல்லையெனில் else பகுதியில் உள்ள கூற்று\_தொகுதி n நிறைவேற்றும்.

**Example:**

```
a=int(input("Enter a value:"))
b=int(input("Enter b value:"))
c=int(input("Enter c value:"))
if a>b and a>c:
    print("A is Big")
elif b>a and b>c:
    print("B is Big")
else:
    print("C is Big")
```

**Output:**

Enter a value:35  
Enter b value:43  
Enter c value:12  
B is Big

**3. அனைத்து மூன்று இலக்க ஒற்றைப்படை எண்களை வெளியிடுவதற்கான நிரலை எழுதுக.**

\*\*[March 2020]

**Program:**

```
for i in range(101,1000,2):
    print(i,end='\t')
```

**Output:**

101 103 105 107 109 111 113 115 ..... 995 997 999

**4. கொடுக்கப்பட்ட எண்ணின் பெருக்கல் வாய்ப்பாட்டை வெளியிடும் நிரலை எழுதுக.**

**Program:**

```
a=int(input("Enter Table which you want: "))
for i in range(1,11,1):
print(i, 'X', a, '=', i*a)
```

**Output:**

```
Enter Table which you want: 5
1 X 5 = 5
2 X 5 = 10
3 X 5 = 15
4 X 5 = 20
5 X 5 = 25
6 X 5 = 30
7 X 5 = 35
8 X 5 = 40
9 X 5 = 45
10 X 5 = 50
```

**1 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)**

- கட்டுப்பாட்டு நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு தாவுவதற்கு காரணமாக நிரல் கூற்றுகள் \_\_\_ எனப்படும்.  
அ)கட்டளை அமைப்பு ஆ)கட்டுப்பாட்டு கட்டமைப்பு  
இ)கட்டுப்பாட்டு கட்டளை ஈ)கூற்று அமைப்பு
- நிகழ்நேர நிரல்களை இயக்கும் போது நிரலின் ஒரு பகுதியை அல்லது கூற்றுகளின் தொகுதியை நிறைவேற்றாமல் விட்டு விட்டு நிபந்தனையின் அடிப்படையில் நிரலின் மற்றொரு பகுதியை இயக்க நேரிடும். இதற்கு \_\_\_ என்று பெயர்.  
அ)மடக்கு ஆ)பன்முறைச்செயல் இ)தாவுதல் ஈ)கிளைப்பிரிப்பு
- கூற்றுகளின் உள்ள ஒரு தொகுதியை பல தடவை நிறைவேற்ற வேண்டி இருக்கும். இதற்கு \_\_\_ என்று பெயர்  
அ)மடக்கு ஆ)பன்முறைச்செயல் இ)தாவுதல் ஈ)அ அல்லது ஆ
- பைத்தானில் \_\_\_ வகையான கட்டுப்பாட்டு அமைப்புகள் உள்ளன  
அ)1 ஆ)2 இ)3 ஈ)4
- ஒன்றன்பின் ஒன்றாக நிறைவேற்றப்படும் கூற்றுகளின் வரிசையைக் கொண்டது \_\_\_ கூற்று ஆகும்  
அ)கிளைப்பிரிப்பு ஆ)மடக்கு இ)வரிசைமுறை ஈ)அனைத்தும்
- பின்வருபவைகளில் எது அனைத்து தீர்மானிப்புக் கூற்றுகளிலும் மிக எளிதான கூற்றாகும்  
அ)if..else ஆ) if..elif இ)எளிய if ஈ)ஏதுமில்லை
- \_\_\_ கூற்றானது சரி தொகுதி மற்றும் தவறு தொகுதி இரண்மையுமே சரிபார்ப்பதற்கான கட்டுப்பாட்டை வழங்குகிறது  
அ)if..else ஆ) if..elif இ)எளிய if ஈ)ஏதுமில்லை
- \_\_\_ பகுதி if..else..elif கூற்றுகளை இணைக்கிறது  
அ)else ஆ) if..elif இ)எளிய if ஈ)ஏதுமில்லை
- C, C++ போன்று பைத்தானில் \_\_\_ ஆனது தொகுதி மற்றும் துணைத் தொகுதியை நெளிவு அடைப்புக்குறிகள் இல்லாமல் ({} )உருவாக்க பயன்படுகிறது  
அ)கிளைப்பிரிப்பு ஆ)வரிசைமுறை இ)உ எத்தள்ளல் ஈ)அனைத்தும்
- \_\_\_ என்பது பயனர் விரும்பும் குறிமுறைத் தொகுதியை குறிப்பிட்ட எண்ணிக்கை வரை அல்லது நிபந்தனை நிறைவேற்றப்படும் வரை இயக்குவதாகும்.  
அ)கிளைப்பிரிப்பு ஆ)வரிசை இ)மடக்கு ஈ)அனைத்தும்



**உதாரணம்:**

```
a=int(input("Enter a number:"))
if a%2==0:
print("The given number is Even")
```

**வெளியீடு:**

```
Enter a number:2
The given number is Even
```

**2. பின்வரும் நிரலுக்கான வெளியீட்டை எழுதுக.**

```
a = int(input("Enter any number:"))
x="even" if a%2==0 else "odd"
print (a, "is", x)
```

**Output:**

```
Enter any number: 7
7 is odd
```

**3. பின்னலான மடக்கு என்றால் என்ன?**

ஒரு மடக்கின் உள்ளே மற்றொரு மடக்கு இடம் பெற்றிருந்தால் அது பின்னலான மடக்கு ஆகும்

**4. பைத்தானில் தாவும் கூற்றுகள் என்றால் என்ன?**

பைத்தானிலுள்ள தாவும் கூற்றுகள் என்பது கட்டுப்பாட்டை எந்தவொரு நிபந்தனையுமின்றி, நிரலின் ஒரு பகுதியிலிருந்து மற்றொரு பகுதிக்கு இடமாற்றம் செய்ய பயன்படுகிறது

**5. break கூற்று என்றால் என்ன?**

**break** கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது. நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது.

**6. pass கூற்று என்றால் என்ன?**

**pass** கூற்று ஒரு null கூற்றாகும். இதை நிறைவேற்றும் போது அதே கூற்றை முழுவதுமாக புறக்கணித்து விடும். இதை இயக்கும் போது எந்த செயல்பாடும் நடைபெறாது

**7. continue கூற்று என்றால் என்ன?**

**continue** கூற்றானது மடக்கின் மீதமுள்ள குறிமுறையைத் தவிர்த்து அடுத்த மடக்கு செயலை ஆரம்பிக்கும்

**3 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)**

**1. கிளைப்பிரித்தல் கூற்றுகளை பட்டியலிடுக.**

பைத்தானில் கீழ்க்கண்ட கிளைப்பிரித்தல் கூற்று வகைகள் உள்ளன .

- எளிய if கூற்று
- if..else கூற்று
- if..elif கூற்று

**2. பைத்தானில் மடக்கு என்றால் என்ன? அதன் வகைகளை எழுதுக.**

பைத்தானில் மடக்கு என்பது பயனர் விரும்பும் குறிமுறைத் தொகுதியை குறிப்பிட்ட எண்ணிக்கை வரை அல்லது நிபந்தனை நிறைவேற்றப்படும் வரை இயக்குவதாகும்.

பைத்தான் இரண்டு வகையான மடக்கு அமைப்புகளை வழங்குகிறது. அவை முறையே,

- while மடக்கு
- for மடக்கு

3. for மடக்கின் பொதுவடிவத்தை தருக.

**for** தொடரியல்\_மாறி **in** வரிசை:

கூற்று\_தொகுதி 1

[else: #optional block

கூற்று\_தொகுதி 2 ]

4. பல்வேறு வகையான தாவும் கூற்றுகளை எழுதுக.

பைத்தானில் மூன்று வகையான தாவும் கூற்றுகள் உள்ளன. அவை முறையே

- break
- continue
- pass

5. பைத்தானில் உள்தள்ளல் என்றால் என்ன?

பைத்தானில் உள்தள்ளல் என்பது மிக முக்கிய பங்கு வகிக்கிறது. தொகுதியை நெளிவு அடைப்புக்குறிகள் ({ }) பயன்படுத்தாமல் கூற்றுகளை ஒரு தொகுதிக்குள் குறிப்பிட உள்தள்ளல் பயன்படுகிறது.

6. பின்வருபவற்றை காண்பிப்பதற்கான நிரல் ஒன்றை எழுதுக

```
*****
****
***
**
*
```

**Program:**

```
for i in range(5):
    for j in range(5-i):
        print('*', end=" ")
    print()
```

**Output:**

```
*****
****
***
**
*
```

5 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

1. கிளைப்பிரித்தல் கூற்று என்றால் என்ன? அதன் பொதுவடிவத்தை தந்து தகுந்த உதாரணங்களை தருக.

நிகழ்நேர நிரல்களை இயக்கும் போது நிரலின் ஒரு பகுதியை அல்லது கூற்றுகளின் தொகுதியை நிறைவேற்றாமல் விட்டு விட்டு நிபந்தனையின் அடிப்படையில் நிரலின் மற்றொரு பகுதியை இயக்க நேரிடும். இதற்கு கிளைப்பிரிப்பு என்று பெயர்.

பைத்தானில் கீழ்க்கண்ட கிளைப்பிரித்தல் கூற்று வகைகள் உள்ளன.

- எளிய if கூற்று
- if..else கூற்று
- if..elif கூற்று

**எளிய if கூற்று:**

**தொடரியல்:**

```
if <நிபந்தனை>:  
    கூற்று_தொகுதி
```

**உதாரணம்:**

```
a=int(input("Enter a number:"))  
if a%2==0:  
    print("The given number is Even")
```

**வெளியீடு**

```
Enter a number:2  
The given number is Even
```

**if else கூற்று:**

**பொதுவடிவம்:**

```
if <நிபந்தனை>:  
    கூற்று_தொகுதி1  
else:  
    கூற்று_தொகுதி2
```

**உதாரணம்:**

```
a=int(input("Enter a number:"))  
if a%2==0:  
    print("The given number is Even")  
else:  
    print("The given number is Odd")
```

**வெளியீடு:**

```
Enter a number:7  
The given number is Odd
```

**if..elif கூற்று:**

if கூற்றுகளைத் தொடர் கூற்றுகளாக அமைக்கும் விரும்பும் போது 'else' பகுதிக்குப் பதிலாக 'elif' பகுதி பயன்படுத்தலாம்.

**தொடரியல்:**

```
if <நிபந்தனை_1>:  
    கூற்று_தொகுதி 1  
elif <நிபந்தனை_2>:  
    கூற்று_தொகுதி 2  
else:  
    கூற்று_தொகுதி n
```

**உதாரணம்:**

```
a=int(input("Enter a value:"))
b=int(input("Enter b value:"))
c=int(input("Enter c value:"))
if a>b and a>c:
    print("A is Big")
elif b>a and b>c:
    print("B is Big")
else:
    print("C is Big")
```

**Output:**

```
Enter a value:35
Enter b value:43
Enter c value:12
B is Big
```

2. பைத்தானில் மடக்கு என்றால் என்ன? அதன் பொதுவடித்தை தந்து பொருத்தமான உதாரணங்களை தருக.

பைத்தானில் மடக்கு என்பது பயனர் விரும்பும் குறிமுறைத் தொகுதியை குறிப்பிட்ட எண்ணிக்கை வரை அல்லது நிபந்தனை நிறைவேற்றப்படும் வரை இயக்குவதாகும்.

பைத்தான் கிரண்டு வகையான மடக்கு அமைப்புகளை வழங்குகிறது. அவை முறையே,

- while மடக்கு
- for மடக்கு

**while மடக்கு:**

பொதுவடிவம்:

```
while <நிபந்தனை>:
    கூற்று_தொகுதி1
[else:                #optional
    கூற்று_தொகுதி2]
```

**Example:**

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

**Output:**

```
1 2 3 4 5
```

**for மடக்கு:**

பொதுவடிவம்:

```
for தொடரியல்_மாறி in வரிசை:
    கூற்று_தொகுதி 1

[else:                #optional block
    கூற்று_தொகுதி 2 ]
```

**உதாரணம்:**

```
for i in range(2,10,2):
    print(i,end=' ')
else:
    print("\nEnd of loop")
```

**வெளியீடு:**

2 4 6 8  
End of loop

3. பைத்தானில் **jump** கூற்றுகள் என்றால் என்ன? அதன் பொதுவடித்தை தந்து பொருத்தமான உதாரணங்களைத் தருக.

பைத்தானிலுள்ள தாவும் கூற்றுகள் என்பது கட்டுப்பாட்டை எந்தவொரு நிபந்தனையுமின்றி, நிரலின் ஒரு பகுதியிலிருந்து மற்றொரு பகுதிக்கு இடமாற்றம் செய்ய பயன்படுகிறது

பைத்தானில் மூன்று வகையான தாவும் கூற்றுகள் உள்ளன. அவை முறையே

- break
- continue
- pass

**break கூற்று:**

**break** கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது. நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது.

**பொதுவடிவம்:**

*break*

**உதாரணம்:**

```
for word in "Jump Statement":
    if word=="e":
        break
    print (word, end=' ')
```

**வெளியீடு:**

Jump Stat

**continue கூற்று:**

**continue** கூற்றானது மடக்கின் மீதமுள்ள குறிமுறையைத் தவிர்த்து அடுத்த மடக்கு செயலை ஆரம்பிக்கும்.

**பொதுவடிவம்:**

*continue*

**உதாரணம்:**

```
for word in "Jump Statement":
    if word == "e":
        continue
    print (word, end=" ")
print ("\n End of the program")
```

**வெளியீடு:**

Jump Statment

End of the program

**pass கூற்று:**

**pass** கூற்று ஒரு null கூற்றாகும். இதை நிறைவேற்றும் போது அதே கூற்றை முழுவதுமாக புறக்கணித்து விடும். இதை இயக்கும் போது எந்த செயல்பாடும் நடைபெறாது.

**பொதுவடிவம்:**

*pass*

**உதாரணம்:**

```
a=int (input("Enter any number :"))
if (a==0):
    pass
else:
    print ("non zero value is accepted")
```

**வெளியீடு:**

Enter any number :3  
 non zero value is accepted

4. பின்வரும் நிரலுக்கான வெளியீட்டை எழுதுக. **\*\*[March 2020]**

**Program:**

```
i=1
while(i<=6):
    for j in range(1,i):
        print(j, end='\t')
    print(end='\n')
    i+=1
```

**Output:**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

### 7. பைத்தான் செயற்கூறுகள்

#### புத்தக வினாக்கள்:

- ஒரு குறிப்பிட்ட செயலை செய்வதற்கான வடிவமைக்கப்பட்டு, பெயரிடப்பட்ட குறிமுறையின் தொகுதி  
அ) மடக்கு                      ஆ)கிளைப்பிரிப்பு                      **இ)செயற்கூறு**                      ஈ)தொகுதி
- தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறை இவ்வாறு அழைப்பர்.  
அ)உள்ளிணைந்த                      **ஆ)தற்கழற்சி**                      இ)லாம்ப்டா                      ஈ)return கூற்று
- எந்த செயற்கூறை பெயரில்லா செயற்கூறு என்று அழைக்கப்படுகிறது?  
**அ) லாம்ப்டா**                      ஆ)தற்கழற்சி                      இ)செயற்கூறு                      ஈ)வரையறை
- செயற்கூறு தொகுதியை எந்த சிறப்புச்சொல் தொடங்கி வைக்கிறது?  
அ)define                      ஆ) for                      இ) finally                      **ஈ) def**
- எந்த சிறப்புச்சொல் செயற்கூறு தொகுதியை முடித்து வைக்கிறது?  
அ)define                      **ஆ) return**                      இ) finally                      ஈ) def
- செயற்கூறு வரையறையில் பின்வரும் எந்த குறியீடு பயன்படுத்தப்படுகிறது?  
அ) ; (அரைப்புள்ளி)                      ஆ) . (புள்ளி)                      **இ) : (முக்காற்புள்ளி)**                      ஈ) \$(டாலர்)
- பின்வரும் கூற்றுகளைப் படித்து, சரியான கூற்றுகளைத் தேர்ந்து எடுக்கவும்.  
(I) பைத்தானில், செயற்கூறை வரையறுக்கும் போது குறிப்பிட்ட தரவு வகைகளைக் குறிப்பிடத் தேவையில்லை  
(II) பைத்தான் சிறப்புச் சொற்களைச் செயற்கூறின் பெயராகப் பயன்படுத்தலாம்.  
**அ) I சரி மற்றும் II தவறு**                      ஆ)இரண்டுமே சரி                      இ)இரண்டுமே தவறு                      ஈ) I தவறு மற்றும் II சரி
- கொடுக்கப்பட்ட கூற்றை வெற்றிகரமாக நிறைவேற்றுவதற்கு, பின்வருவனவற்றுள் சரியான ஒன்றைத் தேர்ந்தெடு .  
  
if \_\_\_\_\_ : print (x, "is a leap year")  
அ) x%2=0                      **ஆ) x%4==0**                      இ) x/4=0                      ஈ) x%4=0
- testpython() செயற்கூறை வரையறுக்க பின்வரும் எந்த சிறப்புச் சொல் பயன்படுகிறது?  
அ) define                      ஆ) pass                      **இ) def**                      ஈ) while

#### 2Marks:

#### 1. செயற்கூறு என்றால் என்ன?

ஒரு குறிப்பிட்ட செயலினை செய்வதற்காக வடிவமைக்கப்பட்டு பெயரிடப்பட்ட குறிமுறையின் தொகுதியே செயற்கூறு எனப்படும்.

#### 2. செயற்கூறின் வகைகளை எழுதுக.

அடிப்படையாக செயற்கூறுகளை பின்வருமாறு வகைப்படுத்தலாம்.

- ◆ பயனர் வரையறுக்கும் செயற்கூறுகள்
- ◆ உள்ளிணைந்த செயற்கூறுகள்
- ◆ லாம்ப்டா செயற்கூறுகள்
- ◆ தற்கழற்சி செயற்கூறுகள்

#### 3. செயற்கூறுவின் முக்கிய நன்மைகள் யாவை?

- ◆ குறிமுறையை மீண்டும் எழுதுவதை தவிர்த்து குறிமுறையின் மறு பயனாக்கத்திற்கு உதவுகிறது.
- ◆ நமது பயன்பாட்டிற்குச் சிறந்த கூறுநிலையை வழங்குகிறது.

4. மாறியின் வரையெல்லை என்றால் என்ன? அதன் வகைகளைக் குறிப்பிடுக.

மாறியின் வரையெல்லை என்பது நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும். அதாவது எந்த பகுதியில் மாறியைப் பயன்படுத்துகிறோமோ அதைக் குறிக்கிறது. வரையெல்லையானது நடப்பு மாறிக் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும்.

இரண்டு வகையான வரையெல்லைகள் உள்ளன.

- 1) உள்ளமை வரையெல்லை
- 2) குளோபல் வரையெல்லை

5. குளோபல் வரையெல்லை-வரையறு.

குளோபல் வரையெல்லை உடைய மாறியை நிரலில் எங்கு வேண்டுமானாலும் அணுக முடியும். எந்த ஒரு செயற்கூறு வரையெல்லைக்கு வெளியேயும் மாறியை வரையறுத்து உருவாக்க முடியும்.

**3Marks:**

1. உள்ளமை மாறிகளுக்கான விதிமுறைகளை எழுதுக.

உள்ளமை மாறியின் விதிமுறைகள்:

- உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.
- செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.
- செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உருவாக்கப்படும்.

2. பைத்தானில் உள்ள முழுதளாவிய சிறப்புச் சொல்லுக்கான அடிப்படை விதிமுறைகளை எழுதுக.

முழுதளாவிய வரையெல்லை சிறப்புச்சொல்லின் விதிமுறைகள்:

- செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக குளோபல் ஆகும். 'global' என்ற சிறப்புச்சொல்லை பயன்படுத்த வேண்டியதில்லை.
- செயற்கூறினுள் முழுதளாவிய மாறியை படிக்க மற்றும் எழுத 'global' சிறப்புச்சொல் பயன்படுத்த வேண்டும்.
- செயற்கூறுவிற்கு வெளியே 'global' என்ற சிறப்புச்சொல் எந்த விளைவையும் ஏற்படுத்தாது.

3. செயற்கூறினுள் முழுதளாவிய மாறியை மாற்றம் செய்தால் என்ன நிகழும்?

செயற்கூறினுள் முழுதளாவிய மாறியை மாற்றம் செய்தால் பின்வருமாறு மாற்றம் நடைபெறும்.

```
x=0 # global variable
def add():
    global x
    x=x+5 #increment by 2
    print("Inside add() function x value is: ",x)
add()
print("In main x values is:",x)
```

**Output:**

```
Inside add() function x value is:5
In main x value is:5
```

4. கொடுக்கப்பட்ட வருடம் லீப் வருடமா இல்லையா என்பதனைச் சோதிக்கும் பைத்தான் நிரலை எழுதுக.

**Program:**

```
year=int(input("Enter year:"))
if (year%4==0):
    print(year,"is leap year")
else:
    print(year,"is not leap year")
```

**Output:**

Enter year:2003  
2003 is not leap year

**5. செயற்கூறில் தொகுப்பு என்பது என்ன?**

செயற்கூறு திருப்பியனுப்பும் மற்றொரு செயற்கூறிற்கு செயலுருபாக, பின்னலான அமைப்பில் பயன்படுத்தினால் அதற்கு தொகுப்பு என்று பெயர்.

**6. செயற்கூறினை வரையறுக்கும் போது குறிப்பிடப்பட வேண்டிய குறிப்புகள் யாவை?**

- பயனர்கள் வரையறுக்கும் செயற்கூறுகள் என்பது பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள் என்பதாகும்.
- செயற்கூறு தொகுதி **def** என்ற சிறப்புச்சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் () அடைப்புக்குறியுடன் முடிய வேண்டும்.
- ஏதேனும் உள்ளீட்டு செயலுருப்புக்கள் அல்லது அளபுருக்கள் இருப்பின் அவற்றை செயற்கூற்றை வரையறுக்கும் போதே () என்ற அடைப்புக்குறிக்குள் கொடுக்க வேண்டும்.
- குறிமுறை தொகுதியானது எப்பொழுதும் முக்காற்புள்ளிக்கு பிறகு உள்தள்ளி வர வேண்டும்.
- **return** கூற்று செயற்கூறினை முடித்து வைக்கும்.

**5Marks:**

**1. செயற்கூறின் வகைகளை எடுத்துக்காட்டுடன் விவரி.**

அடிப்படையாக செயற்கூறுகளை பின்வருமாறு வகைப்படுத்தலாம்.

- ◆ பயனர் வரையறுக்கும் செயற்கூறுகள்
- ◆ உள்ளிணைந்த செயற்கூறுகள்
- ◆ லாம்டா செயற்கூறுகள்
- ◆ தற்கழற்சி செயற்கூறுகள்

**1)பயனர் வரையறுக்கும் செயற்கூறுகள்:**

பயனர்கள் வரையறுக்கும் செயற்கூறுகள் என்பது பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள் ஆகும்.

**2)உள்ளிணைந்த செயற்கூறுகள்:**

உள்ளிணைந்த செயற்கூறுகள் என்பது பைத்தானில் உள்ளடக்கப்பட்ட செயற்கூறுகள் என்பதாகும்.

**3)லாம்டா செயற்கூறுகள்:**

பைத்தானில் பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவிற்கு பெயரில்லாத செயற்கூறு என்று பெயர். இது லாம்டா செயற்கூறுகள் என்றும் அழைக்கப்படும்.

**4)தற்கழற்சி செயற்கூறு:**

தற்கழற்சி செயற்கூறு என்பது தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறுகள் ஆகும்.

**2. மாறியின் வரையெல்லைகளை எடுத்துக்காட்டுடன் விளக்குக.**

மாறியின் வரையெல்லை என்பது நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும். அதாவது எந்த பகுதியில் மாறியைப் பயன்படுத்துகிறோமோ அதைக் குறிக்கிறது. வரையெல்லையானது நடப்பு மாறியைத் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும்.

**இரண்டு வகையான வரையெல்லைகள் உள்ளன.**

- 1) உள்ளமை வரையெல்லை
- 2) குளோபல் வரையெல்லை

**1) உள்ளமை வரையெல்லை:**

ஒரு செயற்கூறுவின் உடற்பகுதியின் உள்ளே அல்லது உள்ளமை வரையெல்லையில் மாறியை அறிவிப்பது உள்ளமை மாறி எனப்படும்.

**உள்ளமை மாறியின் விதிமுறைகள்:**

- உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.
- செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.
- செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உருவாக்கப்படும்.

**உதாரணம்:**

```
def loc():
    y=0          # local
    print(y)
loc()
வெளியீடு:
0
```

**2) குளோபல் வரையெல்லை:**

குளோபல் வரையெல்லை உடைய மாறியை நிரலில் எங்கு வேண்டுமானாலும் அணுக முடியும். எந்த ஒரு செயற்கூறு வரையெல்லைக்கு வெளியேயும் மாறியை வரையறுத்து உருவாக்க முடியும்.

**குளோபல் வரையெல்லை சிறப்புச்சொல்லின் விதிமுறைகள்:**

- செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக குளோபல் ஆகும். 'global' என்ற சிறப்புச்சொல்லை பயன்படுத்த வேண்டியதில்லை.
- செயற்கூறினுள் முழுதளவிய மாறியை படிக்க மற்றும் எழுத 'global' சிறப்புச்சொல் பயன்படுத்த வேண்டும்.
- செயற்கூறுவிற்கு வெளியே 'global' என்ற சிறப்புச்சொல் எந்த விளைவையும் ஏற்படுத்தாது.

**உதாரணம்:**

```
c=1          #global
def add():
    print(c)
    add()
வெளியீடு:
1
```

3. கிரண்டு எண்களின் LCM கண்டுபிடிப்பதற்கான பைத்தான் நிரலை எழுதுக.

**Program:**

```
def lcm(x,y):
    if x>y:
        greater=x
    else:
        greater=y
    while(True):
        if(greater%x==0) and (greater%y==0):
            lcm=greater
            break
        greater+=1
    return lcm
```

```
num1=int(input("Enter first number:"))
num2=int(input("Enter second number:"))
print("The L.C.M of",num1, "and", num2, "is", lcm(num1,num2))
```

**Output:**

```
Enter first number:10
Enter second number:5
The L.C.M of 10 and 5 is 10
```

**1 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)**

- \_\_\_ என்பது ஒரு குறிப்பிட்ட செயலை செய்வதற்கான தொடர்புடைய கூற்றுகளின் தொகுதி ஆகும்  
அ)செயற்குறி                      ஆ)செயலேற்பி                      **கி)செயற்கூறு**                      ஈ)அணிகள்
- பின்வருபவைகளில் எது மறுபயனாக்கத்திற்கும், சிறந்த கூறுநிலையை வழங்க உதவுகிறது?  
அ)செயற்குறி                      ஆ)செயலேற்பி                      **கி)செயற்கூறு**                      ஈ)அணிகள்
- எது பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள் என்று அழைக்கப்படுகிறது?  
அ)லாம்ப்டா                      ஆ)உள்ளிணைந்த                      கி)தற்குழற்சி                      **ஈ)பயனர் வரையறுக்கும்**
- எது பைத்தானில் உள்ளடக்கப்பட்ட செயற்கூறுகள் என்று அழைக்கப்படுகிறது?  
அ)லாம்ப்டா                      **ஆ)உள்ளிணைந்த**                      கி)தற்குழற்சி                      ஈ)பயனர் வரையறுக்கும்
- எது பெயரில்லாத செயற்கூறுகள் என்று அழைக்கப்படுகிறது?  
**அ)லாம்ப்டா**                      ஆ)உள்ளிணைந்த                      கி)தற்குழற்சி                      ஈ)பயனர் வரையறுக்கும்
- எது தற்குழற்சியாக தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறுகள் ஆகும்?  
அ)லாம்ப்டா                      ஆ)உள்ளிணைந்த                      **கி)தற்குழற்சி**                      ஈ)பயனர் வரையறுக்கும்
- செயற்கூறு தொகுதி \_\_\_ என்ற சொல்லுடன் தொடங்கும்  
அ)definition                      ஆ) function                      கி) fun                      **ஈ) def**
- செயற்கூற்றை வரையறுக்கும்போதே \_\_\_ என்ற அடைப்புக்குறிக்குள் கொடுக்க வேண்டும்  
அ) [ ]                      ஆ) { }                      **கி) ( )**                      ஈ) < >
- குறிமுறை தொகுதியானது எப்பொழுதும் \_\_\_ பிறகு உள்தள்ளி வர வேண்டும்  
அ)காற்புள்ளி(,)                      ஆ)அரைப்புள்ளி(;)                      **கி)முக்காற்புள்ளி(:)**                      ஈ)புள்ளி(.)
- ஒரு தொகுதி மற்றொரு தொகுதியைக் கொண்டிருந்தால் அதற்கு \_\_\_ என்று பெயர்  
அ)செயற்கூறு                      ஆ)குறிமுறை                      கி)நிரல்                      **ஈ)பின்னலான தொகுதி**
- பைத்தானில் பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறு \_\_\_ செயற்கூறு எனப்படும்  
**அ)லாம்ப்டா**                      ஆ)உள்ளிணைந்த                      கி)தற்குழற்சி                      ஈ)பயனர் வரையறுக்கும்



### 3. பின்னலான தொகுதி என்றால் என்ன?

ஒரு தொகுதி மற்றொரு தொகுதியைக் கொண்டிருந்தால் அது பின்னலான தொகுதி எனப்படும்.

### 4. பெயரில்லாத செயற்கூறு என்றால் என்ன?

பைத்தானில் பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவிற்கு பெயரில்லாத செயற்கூறு என்று பெயர். பைத்தானில் பெயரில்லாத செயற்கூறுகள் லாம்ப்டா சிறப்புச்சொல்லுடன் வரையறுக்கப்படுகிறது. இது லாம்ப்டா செயற்கூறுகள் என்றும் அழைக்கலாம்.

### 5. லாம்ப்டா அல்லது பெயரில்லா செயற்கூறின் பயன்கள் யாவை?

- லாம்ப்டா செயற்கூறு பெரும்பாலும் சிறிய மற்றும் ஒரு முறை பெயரில்லாத செயற்கூறை உருவாக்க பயன்படுகிறது.
- *filter()*, *map()* மற்றும் *reduce()* போன்ற செயற்கூறுகளுடன் சேர்த்து லாம்ப்டா செயற்கூறுகளைப் பயன்படுத்தலாம்.

### 6. பெயரில்லா செயற்கூறின் பொது வடிவத்தை தருக.

பொதுவடிவம்:

$\lambda$  [ arguments(s) ]: expression

#### உதாரணம்:

```
sum=lambda arg1, arg2: arg1+arg2
print("The sum is:", sum(30,40))
print("The sum is:", sum(-30,40))
```

#### வெளியீடு:

```
The sum is:70
The sum is:10
```

### 3மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

#### 1. பயனர் வரையறுக்கும் செயற்கூறின் நன்மைகள் யாவை?

- செயற்கூறுகள் ஒரு நிரலை சிறு தொகுதியாக பிரிக்க உதவுகிறது. இது குறிமுறையை எளிதாக கையாள உதவுகிறது.
- குறிமுறையின் மறுபயனாக்கத்திற்கு உதவுகிறது.
- செயற்கூறின் செயல்பாடுகளை மாற்றம் செய்வது எளிதாகிறது.

#### 2. return கூற்று பற்றி குறிப்பு எழுதுக.

- **return** கூற்று செயற்கூறினை முடித்து வைத்து அழைப்புக்கூற்றுக்கு மதிப்பை திருப்பி அனுப்பும்.
- ஒரு செயற்கூறு செயற்கூறு மதிப்பை அழைப்புக்கூற்றுக்கு திருப்பியனுப்ப தயாராக இருக்கும்போது **return** கூற்று பயன்படுத்தப்படுகிறது. எனவே, இயக்க நேரத்தில் பல **return** கூற்றுகள் இருந்தாலும் ஒரே ஒரு **return** கூற்று மட்டுமே இயக்கப்படும்.
- செயற்கூறில் ஒன்றுக்கு மேற்பட்ட **return** கூற்றுகள் இருந்தாலும் ஒரே ஒரு **return** கூற்று மட்டுமே இயக்க நேரத்தில் இயக்கப்படும்.

8. சரங்கள் மற்றும் சரங்களை கையாளுதல்

**புத்தக வினாக்கள்:**

- பின்வரும் குறியீடுக்கான வெளியீடு யாது?  
`str="Chennai Schools"`  
`str1[7]="-"`  
 அ) Chennai-Schools    ஆ) Chenna-School    இ) **Type error**    ஈ) Chennai
- பின்வருவனவற்றுள் எது சரங்களை இணைக்க பயன்படும் செயற்குறியாகும்?  
 அ) +    ஆ) &    இ) \*    ஈ) =
- முன்று மேற்கோள் குறிகளுக்குள் தரப்படும் சரமானது பின்வருபவனற்றுள் எதை உருவாக்க அனுமதிக்கும்.  
 அ) ஒரு வரி சரம் **ஆ) பல வரி சரங்கள்**    இ) இரு வரி சரம்    ஈ) ஒன்றுக்கு மேற்பட்ட சரங்கள்
- பைத்தானில் சரங்களானது:  
 அ) மாற்றக்கூடியது **ஆ) மாறாதத்தன்மையுடையது**    இ) பரஸ்பரதன்மையற்றது    ஈ) நெகிழ்வானது
- பின்வருவனவற்றுள் எது சரத்தினை துண்டாக்கும் (Slicing) செயற்குறியாகும்?  
 அ) {}    ஆ) []    இ) < >    ஈ) ( )
- சரத்தின் கீழ் ஒட்டானானது:  
 அ) நேர்மறை எண்கள்    ஆ) எதிர்மறை எண்கள்    இ) அ மற்றும் ஆ    ஈ) **அ அல்லது ஆ**

**2Marks:**

- சரம் என்றால் என்ன?  
 குறியீடுக்களின் அணியை கையாள்வதற்கான ஒரு தரவினம் சரம் ஆகும். சரங்கள் என்பன ஒற்றை, இரட்டை அல்லது முன்று மேற்கோள் குறிகளுக்குள் கொடுக்கப்படும் எழுத்து, எண்கள் அல்லது குறியீடுகளின் சேர்க்கையை கொண்டிருக்கும் ஒருங்கமைவு குறியீடுக்களின் வரிசை ஆகும்.
- பைத்தானில் சரங்களை மாற்றம் செய்யும் முடியுமா?  
 பைத்தானில் சரங்களை மாற்றம் செய்யும் முடியாது. அதாவது சரத்தரவு வகையை வரையறுத்த பின்பு இயங்கு நேரத்தில் அதை மாற்றியமைக்க முடியாது.
- பைத்தானில் சரத்தை எவ்வாறு நீக்குவாய்?  
 பைத்தானில் **del** கட்டளையை பயன்படுத்தி ஒரு முழு சர மாறியையும் நீக்க முடியும்.

- பைத்தான் குறிமுறையின் வெளியீடு யாது?

```
str1="school"
print(str1*3)
```

**Output:**

SchoolSchoolSchool

- சரத்தை துண்டாக்குதல் / பிரித்தல் என்றால் என்ன?

மூலச்சரத்திலிருந்து உள்ள ஒரு துணைச்சரம் துண்டு எனப்படும். மூலச் சரத்திலிருந்து [ ] என்ற செயற்குறி மற்றும் சுட்டு அல்லது கீழ்ஒட்டு மதிப்புகளைக் கொண்டிரு துணைச்சரம் உருவாக்கப்படும்.

**3Marks:**

- கொடுக்கப்பட்ட வடிவத்தை அச்சிடும் பைத்தான் நிரலை எழுதுக.

C O M P U T E R

C O M P U T E  
C O M P U T  
C O M P U  
C O M P  
C O M  
C O  
C

**Program:**

```
string1=input("Enter a string:")
string2=''
index=len(string1)
for i in string1:
    string2=string1[0:index]
    index-=1
    print(string2)
```

2. கீழே கொடுக்கப்பட்டுள்ள பைத்தான் நிரலின் வெளியீடு யாது?

```
str1="welcome"
str2="to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str)
```

**Output:**

weol

**5Marks:**

4. பைத்தானில் பயன்படும் சர செயற்குறிகளை தகுந்த எடுத்துகாட்டுடன் விளக்குக.

பைத்தானில் உள்ள சர செயற்குறிகள் பின்வருமாறு:

- 1) இணைப்பு(+)
- 2) சேர்த்தல் (+=)
- 3) பலமுறை (\*)
- 4) சரத்தை துண்டாக்குதல்
- 5) முன்றாம் அளபுருவைப் பயன்படுத்தி சரத்தை பிரித்தல்

1) இணைப்பு (+) :

இரண்டு அல்லது அதற்கு மேற்பட்ட சரங்களை இணைக்கும் செயல்பாடு சேர்த்தல் எனப்படும். கூட்டல்(+) செயற்குறியானது சரங்களை பைத்தானில் இணைத்துக் கொள்ளப் பயன்படுகிறது.

**உதாரணம்:**

"welcome" + "Python"

வெளியீடு:

welcomePython

2) சேர்த்தல் (+=) :





(b)print(str[0:7])

**Output:**

COMPUTE

5 மதிப்பெண் வினாக்கள்: (கூடுதல் வினாக்கள்)

1. பின்வரும் பைத்தான் குறிமுறைக்கான வெளியீட்டை எழுதுக. **\*\*(March 2020)**

**str1="Welcome to Python"**

(i)print(str1)

(i)print(str1)

**Output:**

Welcome to Python

(ii)print(str1[11:17])

(ii)print(str1[11:17])

**Output:**

Python

(iii)print(str1[11:17:2])

(iii)print(str1[11:17:2])

**Output:**

Pto

(iv)print(str1[::4])

(iv)print(str1[::4])

**Output:**

Wotyn

(v)print(str1[::-4])

(v)print(str1[::-4])

**Output:**

nytoW



- ◆ பைத்தான் List ன் கடைசி உறுப்பிற்கு -1 என்ற சுட்டெண் மதிப்புகளை இருத்துகிறது.
- ◆ இது பின்னோக்கு சுட்டு என்றழைக்கப்படுகிறது.

**உதாரணம்:**

Marks = [10, 20, 30, 40]

Marks	10	20	30	40
நேர்மறை சுட்டெண்	0	1	2	3
எதிர்மறை சுட்டெண்	-4	-3	-2	-1

3. பின்வரும் பைத்தான் குறிமுறையில் x ன் மதிப்பு என்ன?

List1 = [2,4,6[1,3,5]]

x = len(List1)

**வெளியீடு:**

4

4. List ன் del மற்றும் remove() செயற்கூறின் வேறுபாடுகள் யாவை?

del	remove()
del கூற்று தெரிந்த உறுப்புகளை List யிலிருந்து நீக்குவதற்கு பயன்படுகிறது	remove () செயற்கூறு தெரியாத உறுப்புகளை List யிலிருந்து நீக்குவதற்கு பயன்படுகிறது

5. ஒரு Tuples n எண்ணிக்கை உறுப்புகளுடன் உருவாக்குதல்களான தொடரியலை எழுதுக.

**தொடரியல்:**

n எண்ணிக்கை உறுப்புகளுடன் Tuples

**Tuples\_Name = (E1, E2, E3 ..... En)**

அடைப்புக்குறி இல்லாத Tuples உறுப்புகள்

**Tuples\_Name = E1, E2, E3 ..... En**

6. பைத்தானில் set என்றால் என்ன?

பைத்தானில் set என்பது தரவின தொகுப்பின் மற்றொரு வகையாகும். set என்பது மாறக்கூடிய, மற்றும் நகல்கள் இல்லாத வரிசைப்படுத்தப்படாத உறுப்புகளின் தொகுப்பாகும்.

**3Marks:**

1. List ஐ விட மேலான Tuples ன் நன்மைகளை எழுதுக.

- 1) List ன் உறுப்புகள் மாற்றலாம் ஆனால் Tuples ன் உறுப்புகளை மாற்ற முடியாது. இதுவே List மற்றும் Tuples க்கு இடையே உள்ள முக்கியமான வேறுபாடு ஆகும்.
- 2) List ன் உறுப்புகள் சதுர அடைப்புக்குறிக்குள் அடைக்கப்பட்டிருக்கும். ஆனால் Tuple ன் உறுப்புகள் வளைவு குறிக்குள் அடைக்கப்பட்டிருக்கும்.
- 3) Tuples ன் மடக்கு List ஐ காட்டிலும் விரைவானது.

2. sort() பற்றி சிறுகுறிப்பு எழுதுக.

sort() என்பது List ல் உள்ள உறுப்புகளை வரிசையாக்கம் செய்யப் பயன்படும் செயற்கூறு ஆகும்.

**தொடரியல்:**

list.sort(reverse=True/False, key=myfunc)

**உதாரணம்:**

```
mylist=[40,10,70,5]
mylist.sort()
print(mylist)
```

**வெளியீடு:**

[5, 10, 40, 70]

3. பின்வரும் குறிமுறையின் வெளியீடு என்ன?

```
list = [2**x for x in range(5)]
print(list)
```

**வெளியீடு:**

[1, 2, 4, 8, 16]

4. del மற்றும் clear() செயற்கூறுகளுக்கு கிடைப்பையான வேறுபாடுகளை எடுத்துக்காட்டுடன் விளக்குக.

del	clear()
del கூற்று ஆனது முழு List ஐ நீக்குகிறது	clear() செயற்கூறு List ன் அனைத்து உறுப்புகளை நீக்கப் பயன்படுகிறது
<b>தொடரியல்:</b> del list	<b>தொடரியல்:</b> list.clear()
<b>உதாரணம்:</b> mylist = [10,20,30,40] del mylist	<b>உதாரணம்:</b> mylist1 = [50,60,70,80] mylist1.clear()
<b>வெளியீடு</b> List is not defined	<b>வெளியீடு</b> []

5. பைத்தானில் set செயல்பாடுகளை பட்டியலிடுக.

பைத்தானில் set செயல்பாடுகள்:

- ◆ ஒட்டு (Union)
- ◆ வெட்டு (Intersection)
- ◆ வேறுபாடு (Difference)
- ◆ சமச்சீரான வேறுபாடு (Symmetric difference)

**5 Marks:**

1. List -ல் உறுப்பை சேர்ப்பதற்கான பல்வேறு வழிகள் யாவை? பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.

List -ல் ஒரு உறுப்பை சேர்ப்பதற்கு மூன்று வழிமுறைகள் உள்ளன. அவை முறையே,

- 1) append()
- 2) extend()
- 3) insert()

**append():**

append() செயற்கூறானது List ன் கிறுதியில் ஒரே ஒரு உறுப்பை சேரக்கப் பயன்படுகிறது.

தொடரியல்:

list.append [சேரக்கப்பட வேண்டிய உறுப்பு]

உதாரணம்:

```
>>> mylist = [10, 20, 30]
>>> mylist.append (40)
>>> print (mylist)
```

வெளியீடு

[10, 20, 30, 40]

**extend():**

**extend()** செயற்கூறானது List ன் இறுதியில் ஒன்றுக்கு மேற்பட்ட உறுப்புகளை சேரக்கப் பயன்படுகிறது.

தொடரியல்:

list.extend [சேரக்கப்பட வேண்டிய உறுப்பு]

உதாரணம்:

```
>>> mylist.extend [50, 60, 70]
>>> print(mylist)
```

வெளியீடு

[10, 20, 30, 40, 50, 60, 70]

**insert():**

**insert()** செயற்கூறானது List ன் எந்தவொரு இடத்திலும் ஒரு உறுப்பை சேரக்கப் பயன்படுகிறது.

தொடரியல்:

list.insert (சுட்டுண் இடமதிப்பு, உறுப்பு)

உதாரணம்:

```
>>> mylist.insert (2, "Computer")
>>> print(mylist)
```

வெளியீடு

[10, 20, "Computer", 30, 40, 50, 60, 70]

2. range() ன் நோக்கம் என்ன? எடுத்துக்காட்டுடன் விளக்குக.

- ◆ range() என்பது பைத்தானின் தொடர் மதிப்புகளை உருவாக்கப் பயன்படும் செயற்கூறாகும்.
- ◆ range() செயற்கூறை பயன்படுத்தி தொடர் மதிப்புகளுடன் List ஐ உருவாக்கலாம்.
- ◆ range() செயற்கூறு மூன்று செயலுருபுகளைக் கொண்டுள்ளது.

தொடரியல்:

range(start value, end value, step value)

இங்கு,

- ◆ **start value** - என்பது தொடரின் தொடக்க மதிப்பு, சுழியம் தானமைவு தொடக்க மதிப்பாகும்.
- ◆ **end value** - என்பது தொடரின் உச்சவரம்பு, பைத்தான் இறுதி மதிப்பை உச்ச வரம்பு -1 என எடுத்துக் கொள்ளுகிறது.
- ◆ **step value** - என்பது ஒரு விருப்பு செயலுருபு. இது வெவ்வேறு இடைவெளிகளில் மதிப்புகளை உருவாக்கப் பயன்படுகிறது.

**எடுத்துக்காட்டு 1:**

```
for x in range (1, 11):
    print(x)
```

**வெளியீடு:**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

**எடுத்துக்காட்டு 2:**

```
for x in range (2, 11, 2):
    print(x)
```

**வெளியீடு:**

2  
4  
6  
8  
10

3. பின்னலான Tuple என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.

- ◆ பைத்தானில், ஒரு Tuples ஐ மற்றொரு Tuples க்குள் வரையறை செய்வதை **பின்னலான Tuples** என்கிறோம்.
- ◆ பின்னலான tuples ல் ஒவ்வொரு Tuples ல் ஒரு உறுப்பாக கருதப்படுகிறது.
- ◆ for மடக்கு பின்னலான tuples ன் அனைத்து உறுப்புகளை அணுகுவதற்கு பயன்படுகிறது.

**எடுத்துக்காட்டு:**

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5),
           ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
```

```
for i in Toppers:
    print(i)
```

**வெளியீடு:**

('Vinodini', 'XII-F', 98.7)  
( 'Soundarya', 'XII-H', 97.5)  
( 'Tharani', 'XII-F', 95.3)  
( 'Saisri', 'XII-G', 93.8)

4. பைத்தானிலுள்ள பல்வேறு set செயல்பாடுகளை பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.

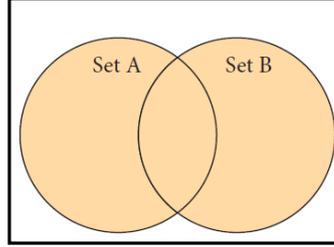
**set செயல்பாடுகள்:**

- ◆ ஒட்டு (Union)
- ◆ வெட்டு (Intersection)

- ◆ வேறுபாடு (Difference)
- ◆ சமச்சீரான வேறுபாடு (Symmetric difference)

### ஒட்டு:

- ◆ இது இரண்டு அல்லது அதற்கு மேற்பட்ட set களின் அனைத்து உறுப்புகளையும் உள்ளடக்கும்.
- ◆ பைத்தானில் | என்ற செயற்குறி இரண்டு set களின் ஒட்டை உருவாக்கப் பயன்படுகிறது.
- ◆ union செயற்கூறும் பைத்தானில் இரண்டு set களை இணைக்கப் பயன்படுகிறது.



### உதாரணம்:

```
set_A={2, 4, 6, 8}
set_B={'A', 'B', 'C', 'D'}
print(set_A | set_B)
```

or

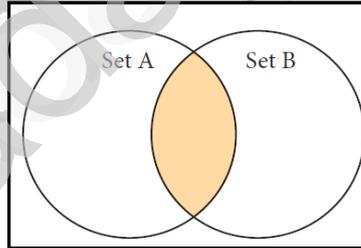
```
set_A={2, 4, 6, 8}
set_B={'A', 'B', 'C', 'D'}
print(set_A.union(set_B))
```

### வெளியீடு:

{2, 4, 6, 8, 'A', 'D', 'C', 'B'}

### வெட்டு:

- ◆ இது இரண்டு set களின் பொதுவான உறுப்புகளை உள்ளடக்கியது.
- ◆ பைத்தானில் & என்ற செயற்குறி இரண்டு set களை வெட்டுகிறது.
- ◆ intersection() செயற்கூறும் பைத்தானில் இரண்டு set களை வெட்டுவதற்கு பயன்படுகிறது.



### உதாரணம்:

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

or

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.intersection(set_B))
```

### வெளியீடு:

{'A', 'D'}

### வேறுபாடு:

- ◆ இது முதல் set(A) ல் உள்ள அனைத்து உறுப்புகளையும் உள்ளடக்கியது.
- ◆ பைத்தானில் - என்ற செயற்குறி set செயற்பாட்டின் வேறுபாட்டைக் கண்டறிய பயன்படுகிறது.





22. பைத்தானில் எந்த செயற்கூறு முழு list யை நீக்க பயன்படுகிறது?  
அ) remove()      ஆ) pop()      இ) del()      ஈ) clear()
23. பைத்தானில் தொடர் மதிப்புகளை உருவாக்க பயன்படும் செயற்கூறு?  
அ) sort()      ஆ) length()      இ) range()      ஈ) reverse()
24. range() செயற்கூறு \_\_\_\_ அளபுருக்களைக் கொண்டது  
அ) 1      ஆ) 2      இ) 3      ஈ) 4
25. பைத்தானில் லிஸ்ட் உருவாக்க பயன்படும் செயற்கூறு எது?  
அ) list()      ஆ) set()      இ) tuples()      ஈ) ஏதுமில்லை
26. \_\_\_\_ என்பது கொடுக்கப்பட்ட நிபந்தனைகளுக்கு உட்பட்டு உருவாகும் தொடர் மதிப்புகளை ஏற்கும் list ஐ உருவாக்கும் எளிய வழிமுறையாகும்  
அ) set சுருக்கம்      ஆ) tuples சுருக்கம்      இ) dictionary சுருக்கம்      ஈ) list சுருக்கம்
27. பைத்தானில் எந்த செயற்கூறு list ன் நகலை தரும்?  
அ) sort()      ஆ) index()      இ) reverse()      ஈ) copy()
28. பைத்தானில் எந்த செயற்கூறு list ல் உள்ள ஒரே மாதிரியான உறுப்புகளின் எண்ணிக்கையை தரும்?  
அ) sort()      ஆ) index()      இ) count()      ஈ) copy()
29. பைத்தானில் எந்த செயற்கூறு list ல் முதலில் வரும் உறுப்பின் சுட்டெண் மதிப்பை தருகிறது?  
அ) sort()      ஆ) index()      இ) count()      ஈ) copy()
30. பைத்தானில் எந்த செயற்கூறு list ல் உள்ள உறுப்புகளின் வரிசையை எண்ணிக்கையை தரும்?  
அ) sort()      ஆ) index()      இ) count()      ஈ) copy()
31. பைத்தானில் எந்த செயற்கூறு list ல் உள்ள உறுப்புகளின் வரிசையை மறுபக்கமாக (தலைகீழாக) திருப்புகிறது?  
அ) sort()      ஆ) index()      இ) reverse()      ஈ) copy()
32. பைத்தானில் எந்த செயற்கூறு list ல் உள்ள உறுப்புகளை வரிசையாக்கம் செய்கிறது?  
அ) sort()      ஆ) index()      இ) reverse()      ஈ) copy()
33. பைத்தானில் எந்த செயற்கூறு list ன் மதிப்புகளில் உச்ச மதிப்பை தரும்?  
அ) min()      ஆ) sum()      இ) max()      ஈ) copy()
34. பைத்தானில் எந்த செயற்கூறு list ல் உள்ள மதிப்புகளில் குறைந்த மதிப்பை தரும்?  
அ) min()      ஆ) sum()      இ) max()      ஈ) copy()
35. பைத்தானில் எந்த செயற்கூறு list டிலுள்ள மதிப்புகளின் கூட்டுத்தொகையை தரும்?  
அ) min()      ஆ) sum()      இ) max()      ஈ) copy()
36. பின்வருபவைகளில் எது காற்புள்ளியால் பிரிக்கப்பட்ட பல மதிப்புகளை வளைவு அடைப்புக் குறிக்குள் கொண்ட தரவினமாகும்?  
அ) Set      ஆ) Tuples      இ) List      ஈ) Dictionary
37. Tuples மாற்ற முடியும். இந்த கூற்று  
அ) சரி      ஆ) தவறு
38. Tuples ல் உள்ள உறுப்புகளை \_\_\_\_ அடைப்புக்குறிக்குள் குறிப்பிட வேண்டும்  
அ) ( )-வளைவு      ஆ) [ ] - சதுர      இ) { } - நெளிவு      ஈ) <> - கோண
39. Tuples வரையறுத்தலின் பின்வருபவைகளில் எந்த கூற்று உண்மையாகும்?  
அ) வளைந்த அடைப்புக்குறிக்குள் வரையறுக்கலாம்      ஆ) வளைந்த அடைப்புக்குறிக்குள் வரையறுக்கலாம்  
இ) அ மற்றும் ஆ இரண்டும்      ஈ) இவற்றில் ஏதுமில்லை
40. பைத்தானில் எந்த செயற்கூறு லிஸ்ட்டிலிருந்து Tuples ஐ உருவாக்க பயன்படுகிறது?





6. List ல் உறுப்புகளை நீக்குவதற்கான பொதுவடிவத்தை எழுதுக.

List ல் உறுப்புகளை நீக்குவதற்கான பொதுவடிவம்:

**தொடரியல்:**

del List [உறுப்பின் சுட்டெண்]

#ஒரு குறிப்பிட்ட உறுப்பை நீக்க

del List [சுட்டெண் முதல் : சுட்டெண் வரை]

#ஒன்றுக்கு மேற்பட்ட உறுப்புகளை நீக்க

del List

#ஒரு List ஐ நீக்க

7. pop() செயற்கூறு என்றால் என்ன?

- pop() செயற்கூறு கொடுக்கப்பட்ட சுட்டெண்ணை பயன்படுத்தி, ஒரு உறுப்பை நீக்கப் பயன்படுகிறது.
- pop() செயற்கூறு சுட்டெண் கொடுக்கப்படாத போது List ன் கடைசி உறுப்பை நீக்கி அதை காண்பிக்கிறது.

8. range() செயற்கூறு என்றால் என்ன?

- ◆ range() என்பது பைத்தானில் தொடர் மதிப்புகளை உருவாக்கப் பயன்படும் செயற்கூறாகும்.
- ◆ range() செயற்கூறைப் பயன்படுத்தி தொடர் மதிப்புகளுடன் List ஐ உருவாக்கலாம்.

9. List சுருக்கம் என்றால் என்ன? அதன் பொதுவடிவத்தை தருக.

List சுருக்கம் என்பது கொடுக்கப்பட்ட நிபந்தனைகளுக்கு உட்பட்டு உருவாக்கும் தொடர் மதிப்புகளை ஏற்கும் List யை உருவாக்கும் எனிய வழிமுறையாகும்.

10. சிறுகுறிப்பு வரைக.

அ) copy()

ஆ) count()

அ) copy()

copy() என்ற செயற்கூறு List ன் நகலை தரும்.

**தொடரியல்:**

list.copy ( )

**எடுத்துக்காட்டு:**

Mylist = [3, 6, 9]

x = Mylist.copy ( )

print(x)

**வெளியீடு:**

[3, 6, 9]

ஆ) count()

count() என்ற செயற்கூறு List ல் உள்ள ஒரே மாதிரியான உறுப்புகளின் எண்ணிக்கையைத் தரும்.

**தொடரியல்:**

list.count (value)

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 6]
x = Mylist.count (6)
print(x)
```

வெளியீடு:

2

11. சிறுகுறிப்பு வரைக.

அ) index()

ஆ) reverse()

அ) index()

index() என்ற செயற்கூறு List ல் முதலில் வரும் உறுப்பின் சுட்டெண் மதிப்பை தருகிறது.

தொடரியல்:

```
list.index (element)
```

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 6]
x = Mylist.index (6)
print(x)
```

வெளியீடு:

1

ஆ) reverse()

reverse() என்ற செயற்கூறு List ல் உள்ள உறுப்புகளின் வரிசையை தலைகீழாக திருப்புகிறது.

தொடரியல்:

```
list.reverse ()
```

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 9]
Mylist.reverse ()
print(Mylist)
```

வெளியீடு:

[9, 6, 3]

12. சிறுகுறிப்பு வரைக.

அ) max()

ஆ) min()

இ) sum()

அ) max()

max() என்ற செயற்கூறு List ன் மதிப்புகளில் உச்ச மதிப்பை தரும்

தொடரியல்:

```
max(List)
```

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 9, 2, 4]  
print (max (Mylist))
```

வெளியீடு:

9

ஆ) min()

min() என்ற செயற்கூறு List ன் மதிப்புகளில் மிகக் குறைந்த மதிப்பைத் தரும்

தொடரியல்:

```
min (List)
```

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 9, 2, 4]  
print (min (Mylist))
```

வெளியீடு:

2

இ) sum()

sum() என்ற செயற்கூறு List ல் உள்ள மதிப்புகளின் கூட்டுத் தொகையை தரும்.

தொடரியல்:

```
sum (List)
```

எடுத்துக்காட்டு:

```
Mylist = [3, 6, 9, 2, 4]  
print (sum (Mylist))
```

வெளியீடு:

24

### 13. Tuples என்றால் என்ன?

Tuples காற்புள்ளியால் பிரிக்கப்பட்ட பல மதிப்புகளை வளைவு அடைப்புக் குறிக்குள் கொண்ட தரவினமாகும். Tuples உள்ள மதிப்புகளை மாற்ற முடியாது.



print(S.\_\_num)

வெளியீடு:

10

Attribute Error

4. பைத்தானில் ஆக்கியை எவ்வாறு உருவாக்குவாய்?

- ஓர் இனக்குழுவின் சான்று பயன்பாட்டிற்கு வரும்பொழுது ஆக்கி என்னும் சிறப்புச் செயற்கூறு தானாகவே இயக்கப்படுகிறது.
- பைத்தானில் "init" என்னும் சிறப்பு செயற்கூறு ஆக்கியாக செயல்படுகிறது.
- இது கிரட்டை அடிக்கீறலில் தொடங்கி கிரட்டை அடிக்கீறலுடன் முடிய வேண்டும்.
- பொருள் உருவாக்கப்பட்டவுடன் ஆக்கியானது தானாகவே இயக்கப்படும்.

5. அழிப்பியின் நோக்கம் என்ன?

- 1) இனக்குழுவில் உருவாக்கப்பட்ட பொருளின் பயன்பாடு முடிவுக்கு வரும் போது அழிப்பி என்னும் சிறப்புச் செயற்கூறு தானாகவே இயக்கப்படும்.
- 2) இது ஆக்கிக்கு முரணானது.
- 3) \_\_del\_\_() செயற்கூறு அழிப்பியாகப் பயன்படுத்தப்படுகிறது.

**3Marks:**

1. இனக்குழு உறுப்புகள் என்றால் என்ன? அதனை எவ்வாறு வரையறுப்பாய்?

- இனக்குழு மாறிகள் மற்றும் வழிமுறைகள் சேர்ந்து இனக்குழுவின் உறுப்புகள் எனப்படும்.
- இனக்குழுவின் உறுப்புகளை இனக்குழுவின் பொருள்கள் அல்லது சான்றுருக்கள் மூலமாகவே அணுகுதல் வேண்டும்.

2. கிரண்டு private இனக்குழு மாறிகளுடன், வழிமுறையைப் பயன்படுத்தி கூட்டுத் தொகை sum அச்சிடும் இனக்குழுவை வரையறுக்கவும்.

```
class sum:
    def __init__(self,n1,n2):
        self.__num1=n1
        self.__num2=n2
    def display(self):
        print(self.__num+self.__num2)
S=Sum(12,14)
S.display()
```

**output:**

26

3. கொடுக்கப்பட்ட வெளியீட்டை பெற விரும்பும் நிரலில் உள்ள பிழைகளை காண்க.

```
class Fruits:
    def __init__(self, f1, f2):
        self.f1=f1
        self.f2=f2
    def display(self):
```

```
print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
```

```
F = Fruits ('Apple', 'Mango')
```

```
del F.display
```

```
F.display()
```

**விடை:**

Fruit 1 = Apple, Fruit 2 = Mango

மேற்கண்ட வெளியீட்டை பெற del F.display என்ற வரியை நிரலில் இருந்து நீக்க வேண்டும்.

4. பின்வரும் நிரலின் வெளியீடு என்ன?

```
class Greeting:
```

```
    def __init__(self, name):
```

```
        self.__name = name
```

```
    def display(self):
```

```
        print("Good Morning ", self.__name)
```

```
obj=Greeting('Bindu Madhavan')
```

```
obj.display()
```

**விடை:**

Good Morning Bindu Madhavan

5. பைத்தானில் ஆக்கி மற்றும் அழிப்புகளை எவ்வாறு வரையறுப்பாய்?

**ஆக்கி:**

```
def __init__(self, [args....]):
```

```
<statement>
```

**அழிப்பி:**

```
def __del__(self):
```

```
<statement>
```

**5 Marks:**

1. எழுது பொருட்களை சேர்க்க அல்லது நீக்கும் பட்டியல் முறை நிரல் ஒன்றை எழுதுக. பொருட்களின் பெயர் மற்றும் பிராண்ட் யை ஒரு dictionary யில் சேமிக்க வேண்டும்.

```
stat = {}
```

```
print("1.Add, 2.Delete, 3.Display, 4.Exit")
```

```
ch = int(input("Enter your choice (1/2/3) : "))
```

```
while (ch!=4):
```

```
    if ch==1:
```

```
        nme=input("Enter stationary name : ")
```

```
        brand=input("Enter brand name : ")
```

```
        stat[nme]=brand
```

```
        pass
```

```
    elif ch==2:
```

```
        nme=input("Enter stationary name to delete : ")
```

```
        del stat[nme]
```

```
    elif ch==3:
```

```
        print(stat)
```

```
else:
    break
ch=int(input("Enter your choice (1/2/3/4) : "))
```

**வெளியீடு:**

```
1.Add, 2.Delete, 3.Display, 4.Exit
Enter your choice (1/2/3) : 1
Enter stationary name : Pen
Enter brand name : Camel
Enter your choice (1/2/3/4) : 1
Enter stationary name : Pencil
Enter brand name : HP
Enter your choice (1/2/3/4) : 1
Enter stationary name : Eraser
Enter brand name : Natraj
Enter your choice (1/2/3/4) : 3
{'Pen': 'Camel', 'Pencil': 'HP', 'Eraser': 'Natraj'}
Enter your choice (1/2/3/4) : 2
Enter stationary name to delete : Pencil
Enter your choice (1/2/3/4) : 3
{'Pen': 'Camel', 'Eraser': 'Natraj'}
Enter your choice (1/2/3/4) : 4
```

**கூடுதல் வினாக்கள்:**

- பைத்தான் ஒரு \_\_\_\_\_ நிரலாக்க மொழி ஆகும்  
அ) கட்டமைப்பு ஆ) இயந்திர **இ) பொருள்நோக்கு** ஈ) இனக்குழு
- பைத்தானில் மிக முக்கிய கட்டமைப்பு கூறுகளாகத் திகழ்பவை \_\_\_\_\_ ஆகும்  
அ) செயற்கூறு ஆ) செயற்குறி **இ) இனக்குழு** ஈ) பொருள்
- \_\_\_\_\_ என்பது பொருளின் வார்ப்புரு ஆகும்  
அ) செயற்கூறு ஆ) செயற்குறி **இ) இனக்குழு** ஈ) பொருள்
- பொருள்நோக்கு நிரலாக்கத்தின்படி \_\_\_\_\_ என்பன இனக்குழுவின் சான்று அல்லது இனக்குழு மாறி என்று அழைக்கப்படும்  
அ) செயற்கூறு ஆ) செயற்குறி **இ) இனக்குழு** **ஈ) பொருள்**
- \_\_\_\_\_ எனபது தரவுகளின் மீது செயல்படும் செயற்கூறு மற்றும் தரவுகளின் தொகுப்பாகும்  
அ) செயற்கூறு ஆ) செயற்குறி **இ) இனக்குழு** **ஈ) பொருள்**
- பைத்தானில், இனக்குழுவை வரையறுக்க \_\_\_\_\_ என்னும் சிறப்புச்சொல் பயன்படுத்தப்படுகிறது  
அ) object **ஆ) class** **இ) constructor** ஈ) destructor
- ஒவ்வொரு இனக்குழுவும் தனித்த பெயருடன் \_\_\_\_\_ யைக் கொண்டு முடியும்.  
அ) காற்புள்ளி(.) ஆ) அரைப்புள்ளி(;) **இ) முக்காற்புள்ளி(:)** ஈ) புள்ளி (.)
- இனக்குழுவுக்குள் வரையறுக்கப்படும் மாறிகள் \_\_\_\_\_ எனப்படும்  
அ) மாறிகள் ஆ) பொருள்கள் **இ) இனக்குழு மாறிகள்** ஈ) வழிமுறைகள்
- இனக்குழுவுக்குள் வரையறுக்கப்படும் செயற்கூறுகள் \_\_\_\_\_ எனப்படும்

- அ) மாறிகள்                      ஆ) பொருள்கள்                      இ) இனக்குழு மாறிகள்                      ஈ) வழிமுறைகள்
10. பின்வருபவைகளில் எது இனக்குழுவின் உறுப்புகள் எனப்படுகின்றன?  
அ) இனக்குழு மாறிகள்                      ஆ) வழிமுறைகள்                      இ) அ மற்றும் ஆ                      ஈ) எதுமில்லை
11. இனக்குழு வழிமுறையில் அளபுருக்களை எடுத்துக் கொள்ளாத போது, முதல் அளபுருவாக \_\_\_ வரையறுக்கப்பட வேண்டும்  
அ) \_\_init\_\_                      ஆ) \_\_del\_\_                      இ) class                      ஈ) self
12. ஓர் இனக்குழுவின் சான்று முடிவுக்கு வரும் பொழுது \_\_\_ என்னும் சிறப்புச் செயற்கூறு தானாகவே இயக்கப்படுகிறது  
அ) அழிப்பி                      ஆ) ஆக்கி                      இ) இனக்குழு                      ஈ) பொருள்
13. ஓர் இனக்குழுவின் சான்று பயன்பாட்டிற்கு வரும்பொழுது \_\_\_ என்னும் சிறப்புச் செயற்கூறு தானாகவே இயக்கப்படுகிறது  
அ) அழிப்பி                      ஆ) ஆக்கி                      இ) இனக்குழு                      ஈ) பொருள்
14. இனக்குழு ஆக்கியின் பெயர் அறிவித்தலில் பெயருக்கு முன்னும் பின்னும் எதை அளிக்க வேண்டும்?  
அ) அடிக்ஈறல்                      ஆ) கிரட்டை அடிக்ஈறல்                      இ) புள்ளி                      ஈ) அரைப்புள்ளி
15. இனக்குழுவின் உள்ளே வரையறுக்கப்படும் மாறி கொடாநிலையாக \_\_\_ ஆகும்  
அ) private                      ஆ) public                      இ) protected                      ஈ) none
16. இனக்குழுவில் private மாறி பயன்படுத்த அந்த மாறியின் பெயருக்கு முன்னொட்டாக \_\_\_ பயன்படுத்த வேண்டும்  
அ) அடிக்ஈறல்                      ஆ) கிரட்டை அடிக்ஈறல்                      இ) புள்ளி                      ஈ) அரைப்புள்ளி

### 2Marks:

1. பொருள் என்றால் என்ன?

பொருள் என்பது தரவுகளின் மீது செயல்படும் செயற்கூறு மற்றும் தரவுகளின் தொகுப்பாகும்.

2. இனக்குழுவின் பொதுவடிவம் (தொடரியலை) தருக.

**தொடரியல்:**

class class\_name:

    கூற்று\_1

    கூற்று\_2

    .....

    .....

    கூற்று\_n

3. இனக்குழு மாறிகள் என்றால் என்ன?

இனக்குழுக்குள்ளே வரையறுக்கப்படும் மாறிகள் இனக்குழு மாறிகள் எனப்படும்.

4. இனக்குழு உறுப்புகளை எவ்வாறு அணுகுவாய்?

இனக்குழு உறுப்புகளை அதாவது இனக்குழு மாறி அல்லது வழிமுறைகளை அணுக புள்ளி (.) செயற்குறி பயன்படுகிறது.

### 5 Marks:

1. பின்வரும் நிரலுக்கான வெளியீட்டைக் காண்க.

class Sample:

```
num=0
def __init__(self, var):
    Sample.num+=1
    self.var=var
    print("The object value is = ", var)
    print("The count of object created = ", Sample.num)
```

```
S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

**output:**

```
The object value is = 15
The count of object created = 1
The object value is = 35
The count of object created = 2
The object value is = 45
The count of object created = 3
```

2. பின்வரும் நிரலுக்கான வெளியீட்டைக் காண்க.

```
class Circle:
    pi=3.14
    def __init__(self,radius):
        self.radius=radius
    def area(self):
        return Circle.pi*(self.radius**2)
    def circumference(self):
        return 2*Circle.pi*self.radius
r=int(input("Enter Radius: "))
C=Circle(r)
print("The Area =",C.area())
print("The Circumference =", C.circumference())
```

**output:**

```
Enter Radius: 5
The Area = 78.5
The Circumference = 31.400000000000002
```

3. பின்வரும் நிரலுக்கான வெளியீட்டைக் காண்க.

```
class String:
    def __init__(self):
        self.uppercase=0
        self.lowercase=0
        self.vowels=0
        self.consonants=0
        self.spaces=0
        self.string=""
```

```
def getstr(self):
```

```

        self.string=str(input("Enter a String: "))
    def count_upper(self):
        for ch in self.string:
            if (ch.isupper()):
                self.uppercase+=1
    def count_lower(self):
        for ch in self.string:
            if (ch.islower()):
                self.lowercase+=1
    def count_vowels(self):
        for ch in self.string:
            if (ch in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u','U')):
                self.vowels+=1
    def count_consonants(self):
        for ch in self.string:
            if (ch not in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u','U')):
                self.consonants+=1
    def count_space(self):
        for ch in self.string:
            if (ch==" "):
                self.spaces+=1
    def execute(self):
        self.count_upper()
        self.count_lower()
        self.count_vowels()
        self.count_consonants()
        self.count_space()
    def display(self):
        print("The given string contains...")
        print("%d Uppercase letters"%self.uppercase)
        print("%d Lowercase letters"%self.lowercase)
        print("%d Vowels"%self.vowels)
        print("%d Consonants"%self.consonants)
        print("%d Spaces"%self.spaces)
    
```

```

S = String()
S.getstr()
S.execute()
S.display()
    
```

**Output:**

```

Enter a String:Welcome To Learn Computer Science
The given string contains...
5 Uppercase letters
24 Lowercase letters
12 Vowels
17 Consonants
3 Spaces
    
```