

அனைத்து வினாக்களுக்கும் விடையளி

1. துணைநிரல் என்றால் என்ன?

- ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக மீண்டும் மீண்டும் பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே துணை நிரலாக்கமாகும்.

2. நிரலாக்க மொழியைப் பொறுத்து செயற்கூறுவை வரையறுக்கவும்.

- செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும்.
- இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும்
- செயற்கூறானது பல வகை உள்ளீடுகளைக் கொண்டு செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.

3. X:=(78) இதன் மூலம் அறிவது என்ன?

- X:=(78), இது ஒரு செயற்கூறு வரையறை ஆகும்.
- இங்கு, 78 என்ற மதிப்பானது X என்ற பெயருடன் பிணைக்கப்படுகிறது.

4. இடைமுகத்தையும், செயல்படுத்துதலையும் வேறுபடுத்துக.

இடைமுகம்	செயல்படுத்துதல்
ஒரு பொருள் செய்யக்கூடிய நடவடிக்கையை வரையறுக்கிறது, ஆனால் அவற்றை உண்மையில் செய்யக் கூடியது இல்லை.	இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.

5. பின்வருவனவற்றுள் எது சாதாரண செயற்கூறு வரையறை மற்றும் எது தற்சுழற்சி செயற்கூறு வரையறை

i) let sum x y:

return x + y

ii) let disp :

print 'welcome'

iii) let rec sum num:

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

i) சாதாரண செயற்கூறு

ii) சாதாரண செயற்கூறு

iii) தற்சுழற்சி செயற்கூறு

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

1. இடைமுகத்தின் பண்புகூறுகள் யாவை?

- ஒரு பொருளை முறையாக உருவாக்கி வழங்குவதற்கும் அதனை செயல்படுத்துவதற்கும் தேவையான இடைமுகத்தை இனக்குழு வார்ப்புரு குறிப்பிடுகிறது.
- செயற்கூறுகளைப்பொருளுக்கு அனுப்புவதன் மூலம் பொருளின் பண்புகளையும் பண்புகூறுகளையும் கட்டுப்படுத்த முடிகிறது.

2. strlen ஏன் pure செயற்கூறு என்று அழைக்கப்படுகிறது?

- strlen என்பது pure செயற்கூறாகும்.
- ஏனென்றால், செயற்கூறு அளபுருவாக ஒரே ஒரு மாறியை எடுத்துக்கொண்டு அதனுடைய நீளத்தை கணக்கிடுகிறது.
- இந்த செயற்கூறு வெளி நினைவகத்தில் இருந்து உள்ளீட்டை எடுத்துக் கொள்கிறது. ஆனால் மதிப்புகளை மாற்றுவதில்லை திருப்பி அனுப்பும் மதிப்புகள் வெளி நினைவகத்தில் இருந்து பெறப்பட்டதாகும்.

3. impure செயற்கூறுவின் பக்க விளைவுகள் யாவை? எடுத்துக்காட்டுடன் விளக்குக?

- செயற்கூறுக்கு அளபுருக்களை அனுப்பாதபோதும், செயற்கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும்.
- ஒரு செயற்கூறு அந்த வரையறை தொகுதியின் வெளியே உள்ள மாறிகள் அல்லது செயற்கூறுகளைச் சார்ந்து இருந்து ஒவ்வொரு முறை அழைக்கும் பொழுதும் செயற்கூறு ஒரே மாதிரியாக இயக்கப்படும் என கூற இயலாது.
- எடுத்துக்காட்டாக random() செயற்கூறு ஒரே மாதிரியான அழைப்புக்கூற்றுக்கு வெவ்வேறு விதமான வெளியீடுகளைக் கொடுக்கும்.

4. pure மற்றும் impure செயற்கூற்றை வேறுபடுத்துக.

Pure செயற்கூறு	Impure செயற்கூறு
Pure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்தே அமையும்.	செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்து அமையாது.
pure செயற்கூற்றினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் எப்பொழுதும் அதே திருப்பி அனுப்பும் மதிப்பே கிடைக்கும்	impure செயற்கூற்றினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் வெவ்வேறான திருப்பி அனுப்பும்
இது எந்த பக்க விளைவுகளையும் கொண்டிருக்காது.	இது பக்க விளைவுகளை கொண்டிருக்கும்
இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யாது.	இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யும்.
எடுத்துக்காட்டு: strlen()	எடுத்துக்காட்டு: random()

5. ஒரு செயற்கூறிக்கு வெளியே ஒரு மாறியை மாற்றினால் என்ன விளைவுகள் ஏற்படும்?

ஒரு எடுத்துக்காட்டு தருக.

- செயற்கூறுவின் வெளியே மாறியை மாற்றம் செய்வது என்பது பக்க விளைவுகளில் ஒன்றாகும்.

```

எடுத்துக்காட்டு      y:=0
                        let inc (x:int):int:=
                            y:=y+x
                            return (y)
    
```

- y-ன் மதிப்பு செயற்கூறு வரையறையின் உள்ளே மாறுவதால் விடையானது ஒவ்வொரு முறையும் மாறும். inc() செயற்கூறுவின் பக்க விளைவு என்னவென்றால் வெளிப்புற மாறியான 'y' ன் மதிப்பை மாற்றுவதாகும்.

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி

1. அளபுருக்கள் என்றால் என்ன?

(அ) தரவுவகை இல்லாத அளபுருக்கள்

(ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் விவரி?

அளபுருக்கள்

- அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும்.

(அ) தரவுவகை இல்லாத அளபுருக்கள்

```

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

- மேலேயுள்ள செயற்கூறு வரையறையில் 'b' என்ற மாறி அளபுரு ஆகும்.
- மாறி 'b'க்கு அனுப்பப்படும் மதிப்பானது செயலுருபு ஆகும்.
- செயற்கூறின் முன் நிபந்தனை (requires) மற்றும் பின் நிபந்தனை (return) கொடுக்கப்பட்டுள்ளது.
- இங்கு நாம் எந்த தர வினத்தையும் குறிப்பிடவில்லை.
- மேலே உள்ள செயற்கூறு வரையறையில், if கோவை, then கிளைக்கு மதிப்பு 1 யைத் திருப்பி அனுப்பினால், தரவு வகை (data type) விதிப்படி if கோவை முழுவதுமே 'int' தரவு வகைக் கொண்டிருக்கும்.
- செயற்கூறின் திருப்பி அனுப்பும் மதிப்பும் int ஆகவே இருக்கும்.

(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 a (b-1)
```

- 'a' மற்றும் 'b' தரவு வகை குறிப்பு எழுதும் போது, அடைப்புக்குறிகள் () அவசியமானது ஆகும்.
- பொதுவாக, இந்த குறிப்புகளை நாம் விட்டுவிடலாம், ஏனெனில், நிரல்பெயர்ப்பி இவற்றை அனுமாணிப்பது மிகவும் எளிது. முன்பெல்லாம் நாம் வெளிப்படையாகவே தரவு வகைகளை எழுதுவோம்.
- எந்த வித அர்த்தமும் இல்லாத தரவு வகை பிழைச் செய்தியைப் பெறும் போது, இது மிகவும் பயனுள்ளதாகும்.

2. பின்வரும் நிரலில்

```
let rec gcd a b :=
```

```
if b <> 0 then gcd b (a mod b) else return a
```

அ) செயற்கூறுவின் பெயர்

ஆ) தற்சுழற்சி செயற்கூறு கூற்று

இ) அளபுருக்கள் கொண்ட மாறியின் பெயர்

ஈ) செயற்கூறுவை தற்சுழற்சிக்கு அழைக்கும் கூற்று

உ) தற்சுழற்சியை முடிவுக்கு கொண்டுவரும் கூற்று ஆகியவற்றை எழுதுக.

அ) gcd

ஆ) let rec gcd a b :=

இ) a b

ஈ) gcd b (a mod b)

உ) return a

3. pure மற்றும் impure செயற்கூறுவை எடுத்துக்காட்டுடன் விளக்குக.

pure செயற்கூறு

- ஒரே மாதிரியான செயலுருபுகளை அனுப்பும் போது, ஒரே மாதிரியான விடையைத் தரும் செயற் கூறு pure செயற் கூறுகள் ஆகும்.
- எடுத்துக்காட்டு:

```
let square x
return: x * x
```
- மேலேயுள்ள square செயற் கூறு pure செயற்கூறு ஆகும்.
- இதன் ஒரு நன்மை என்னவென்றால், pure செயற்கூறாக இருக்கும் பொழுது, அதே அளபுருக்களுடன் செயற்கூறுவை பல தடவைகள் அழைக்கும் போது, உண்மையில் நிரல்பெயர்ப்பி செயற்கூறுவை மீண்டும் ஒரு தடவை அழைக்கும் தேவை மட்டுமே ஏற்படுகிறது.

impure செயற்கூறு

- செயற்கூறுக்கு அளபுருக்களை அனுப்பாத போதும், செயற் கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும். இந்த வகையான செயற்கூறை impure செயற்கூறு என்பர்.
- ஒரு செயற்கூறு வரையறை தொகுதியின் வெளியே உள்ள மாறிகள் அல்லது செயற் கூறுகளைச் சார்ந்து இருந்து ஒவ்வொரு முறை அழைக்கும் பொழுதும் செயற்கூறு ஒரே மாதிரியாக இயக்கப்படும் என கூற இயலாது.

```
let randomnumber:=
```

```
a:=random()
```

```
if a>10 then
```

```
return: a
```

```
else
```

```
return: 10
```

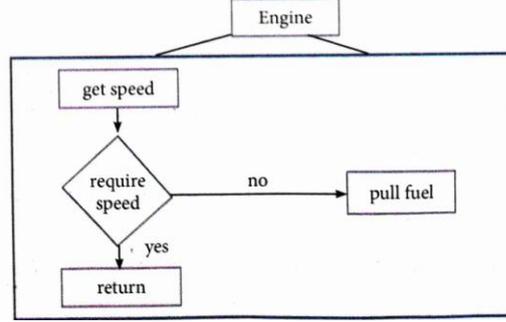
- இங்கு, Random என்பது impure செயற்கூறு ஆகும். ஏனெனில் இதனை அழைக்கும் பொழுது என்ன விடை கிடைக்கும் என நிச்சயமாக கூற இயலாது.

4. இடைமுகம் மற்றும் செயல்படுத்துதலை எடுத்துக்காட்டுடன் விளக்குக.

இடைமுகம்

- ஒரு பொருள் (Object) செய்யக்கூடிய செயல்களின் தொகுப்பு இடைமுகம் ஆகும்.

- பொருள் நோக்கு நிரலாக்க மொழியில், இடைமுகம் என்பது அனைத்து செயற்கூறுகளின் விவரிப்பு ஆகும்.
- மின் விளக்கை போல் செயல்படும் எதுவும் turn_on() மற்றும் turn_off() என்ற செயற்கூறு வரையறையைக் கொண்டிருக்கும்.
- இணக்குமுவிண் பண்புக்கூறுகளை செயல்படுத்த இடைமுகம் பயன்படுகிறது.
- இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.
- பொருள் நோக்கு நிரலாக்க மொழியில் இணக்குமு என்பது இடைமுகம் ஆகும்.



செயல்படுத்துதல்

- செயல்படுத்துதல் என்பது பொருள் எவ்வாறு செயல்படுத்தப்பட்ட இயங்குகிறது என்பதை குறிப்பதாகும்.
- காரை ஓட்டும் நபர் அந்த காரின் உட்புற செயல்பாடுகள் பற்றி அறிந்திருக்க வேண்டிய அவசியமில்லை.
- காரின் வேகத்தை அதிகப்படுத்த, அவர் காரின் துரிதப்படுத்தியை அழுத்தி விரும்பிய பண்பை பெறுவார். இங்கு துரிதப்படுத்தி என்பது கார் ஓட்டுநருக்கும் இயந்திரத்துக்கும் இடையேயான இடைமுகம் ஆகும்.
- இதில், அழைக்கும் செயற்கூறு speed(70): என இருக்க வேண்டும் இது ஒரு இடைமுகமாகும்.

பாடம் - 2 தரவு அருவமாக்கம் பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

1. தரவு அருவமாக்கம் வகை என்றால் என்ன?

- அருவமாக்க தரவு வகை என்பது பொருள்களுக்கான வகை ஆகும், இதன் செயல் மதிப்பின் தொகுப்பு மற்றும் செயல்பாடுகளின் தொகுப்பால் வரையறுக்கப்படுகிறது.

2. ஆக்கிகள் மற்றும் செலக்டர்கள் வேறுபாடு தருக.

ஆக்கிகள்	செலக்டர்கள்
ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது.	செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.

3. Pair என்றால் என்ன? எடுத்துக்காட்டு தருக.

- இரு மதிப்புகளை ஒன்றாக இணைக்கும் முறையை Pairs என்று அழைக்கிறோம்.
- எடுத்துக்காட்டு: list[(0, 10), (1, 20)]

4. List என்றால் என்ன? எடுத்துக்காட்டு தருக

- List அமைப்பு கோவைகளை சதுர அடைப்புக்குறிக்குள் காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும்.
- List பல மதிப்புகளை சேமிக்கும் இம்மதிப்புகள் எவ்வகையாகவும் இருக்கலாம் அல்லது மற்றொரு லிஸ்டாகவும் இருக்கலாம்.
- எடுத்துக்காட்டு: list=[10,20]
x,y=list

5. Tuple என்றால் என்ன? எடுத்துக்காட்டு தருக.

- Tuple என்பது அடைப்புக்குறிக்குள் கொடுக்கப்பட்டுள்ள தொடர் மதிப்புகளை காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும்.
- எடுத்துக்காட்டு: colour= ('red', 'blue', 'Green')

அனைத்து வினாக்களுக்கும் விடையளி

1. காண்கிரிட் தரவு வகை மற்றும் அருவமாக்கம் தரவு வகை வேறுபடுத்துக.

காண்கிரிட் தரவு வகை	அருவமாக்கம் தரவு வகை
காண்கிரிட் தரவு வகையின் உருவமைப்பு அறியப்பட்டது.	அருவமாக்கம் தரவு வகையின் உருவமைப்பு அறியப்படாது.
எளிய கருத்தினை செயல்படுத்த உதவுகிறது	கருத்து பற்றிய உயர் நிலை பார்வை வழங்குகிறது.

2. நிரல் வடிவமைப்பில் பின்பற்றப்படும் யுக்தி எது? யுக்தியை வரையறுக்க.

- நிரல் வடிவமைப்பில் பின்பற்றப்படும் யுக்தி விருப்பார்வ எண்ணம்.
- விருப்பார்வ எண்ணம் என்பது யதார்த்தத்திற்கு பதிலாக நம்பிக்கையின் அடிப்படையில் விருப்பதற்கு ஏற்ப முடிவெடுப்பதாகும்.

3. பின்வருவனவற்றில் எது constructors and selectors என்று அடையாளம் காணவும்?

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

- a) Constructor
 b) Selector
 c) Selector
 d) Selector
 e) Constructor
 f) Selector

4. list உள்ள உருப்புகளை அணுகும் பல்வேறு வழிமுறைகள் யாவை? எடுத்துக்காட்டு தருக.

- list உள்ள உருப்புகளை அணுகும் இரண்டு வழிமுறைகள் உள்ளன. அவை,
 - பன்மடங்கு மதிப்பிருத்தல்
 - உறுப்புகள் தேர்வு செயற்குறிகள்

பன்மடங்கு மதிப்பிருத்தல்

- list-ன் உறுப்புகள் பிரிக்கப்பட்டு, அனைத்து உறுப்புகளும் வேறு பெயர்களுடன் இணைக்கப்படுகிறது.

lst := [10, 20]

x, y := lst

- மேலே காணும் எடுத்துக்காட்டில், xயின் மதிப்பு 10 என்றும், Y யின் மதிப்பு 20 என மதிப்பிருத்தப்படும்.

உறுப்புகள் தேர்வு செயற்குறிகள்

- லிஸ்ட் லிட்டரல்ஸ் போல் அல்லாது கோவையில் ஒரு சதுர அடைப்புக்குறியை தொடர்ந்து வரும் மற்றொரு சதுர அடைப்புக்குறி லிஸ்டின் மதிப்புகளாக எடுத்துக்கொள்ளப்படுவதில்லை.
- ஆனால் முந்தைய கோவையின் உறுப்பை மதிப்பாக தேர்வு செய்யும்.

lst[0]

10

lst[1]

20

- மேலே காணும் எடுத்துக்காட்டினை கணித முறை யில் set போன்று அமைக்கலாம்.

5. பின்வருவனவற்றில் எது List, Tuple மற்றும் இனக்குழு (class) என அடையாளம் காண்க.

- (a) arr [1, 2, 34] (b) arr (1, 2, 34) (c) student [mo, name, mark] (d) day= ('sun', 'mon', 'tue', 'wed')
 (e) x= [2, 5, 6.5, [5, 6], 8.2] (f) employee [eno, ename, esal, eaddress]

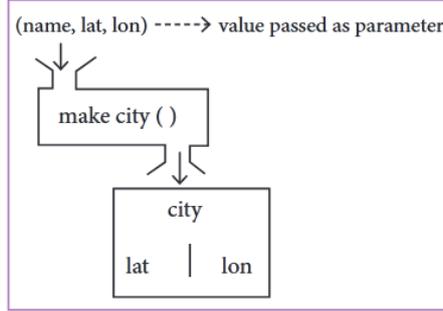
- a) List
 b) Tuple
 c) Class
 d) Tuple
 e) List
 f) Class

அனைத்து வினாக்களுக்கும் விடையளி

1. தரவு அருவமாக்கம் எவ்வாறு செயல்படுத்துவாய்? எடுத்துக்காட்டுடன் விளக்குக.

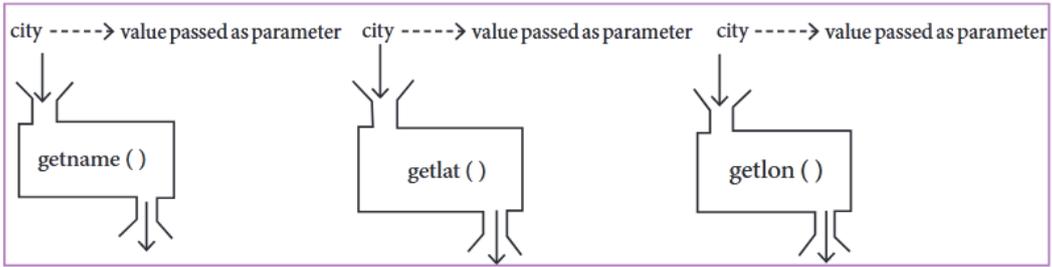
- தரவு அருவமாக்கம் ஆக்கிகள் மற்றும் செலக்டர்கள் கொண்டு செயல்படுத்தலாம்.
ஆக்கி

- ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது.
- எடுத்துக்காட்டாக, city என்று ஒரு அருவமாக்க தரவு வகை உள்ளது என வைத்துக்கொள்.
- City என்ற பொருள் நகரத்தின் பெயர், அட்சரேகை மற்றும் தீர்க்கரேகை பற்றிய தகவல்களை சேமித்திருக்கும் city என்ற பொருளை உருவாக்க பின்வரும் செயற்கூற்றினை பயன்படுத்தலாம்.
city:= makecity (name, lat, lon)
- இங்கு makecity(name, lat, lon) என்ற ஆக்கி city எனும் பொருளை உருவாக்குகிறது.



செலக்டர்

- செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.
- city பொருளின் தகவல்களை பெறுவதற்கு
- பின்வரும் செயற்கூறுகளை பயன்படுத்தலாம்.
 - getname(city)
 - getlat(city)
 - getlon(city)
- மேலே காணும் குறிமுறை செலக்டர் செயற்கூறுகளாகும், இவை city எனும் பொருளிலிருந்து தகவல்களை பெற்றுத்தரும்.



2. List என்றால் என்ன? ஏன் List, Pairs என்று அழைக்கப்படுகிறது. எடுத்துக்காட்டுடன் விவரி.

List

- List அமைப்பு கோவைகளை சதுர அடைப்புக்குறிக்குள் காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும்.
- List பல மதிப்புகளை சேமிக்கும் இம்மதிப்புகள் எவ்வகையாகவும் இருக்கலாம் அல்லது மற்றொரு லிஸ்டாகவும் இருக்கலாம்.
- எடுத்துக்காட்டு

lst:=[10,20]

x,y=lst

பன்மடங்கு மதிப்பிருத்தல்

- list-ன் உறுப்புகள் பிரிக்கப்பட்டு, அனைத்து உறுப்புகளும் வேறு பெயர்களுடன் இணைக்கப்படுகிறது.

lst := [10, 20]

x, y := lst

- மேலே காணும் எடுத்துக்காட்டில், xயின் மதிப்பு 10 என்றும், Y யின் மதிப்பு 20 என மதிப்பிடுதலும்.

உறுப்புகள் தேர்வு செயற்குறிகள்

- லிஸ்ட் லிட்டரல்ஸ் போல் அல்லாது கோவையில் ஒரு சதுர அடைப்புக்குறியை தொடர்ந்து வரும் மற்றொரு சதுர அடைப்புக்குறி லிஸ்டின் மதிப்புகளாக எடுத்துக்கொள்ளப்படுவதில்லை.
- ஆனால் முந்தைய கோவையின் உறுப்பை மதிப்பாக தேர்வு செய்யும்.

```
lst[0]
    10
lst[1]
    20
```

- மேலே காணும் எடுத்துக்காட்டினை கணித முறையில் set போன்று அமைக்கலாம்.
lst[(0, 10), (1, 20)] - where



- இரு மதிப்புகளை ஒன்றாக இணைக்கும் முறையை Pairs என்று அழைக்கிறோம்.
- எடுத்துக்காட்டு

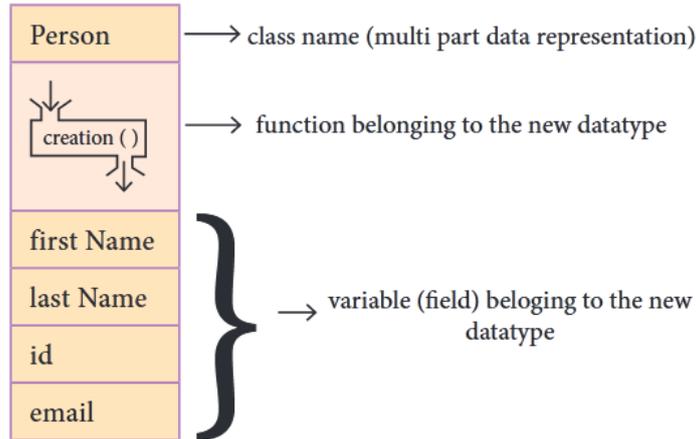
```
lst[(0, 10), (1, 20)]
```

3. பல் உருப்பு பொருளை எவ்வாறு அணுகுவாய் எடுத்துக்காட்டுடன் விளக்கு.

- structure அமைப்பை (OOP மொழியில் இது இனக்குழு அமைப்பு என்று அழைக்கப்படுகிறது) கொண்டு பல உருப்படி பொருளில் அனைத்து உறுப்புகளும் பெயரிடப்படும்.
- பின்வரும் போலிக்குறிமுறையை காணவும்:

```
class Person:
    creation( )
    firstName := " "
    lastName := " "
    id := " "
    email := " "
```

- Person எனும் புதிய தரவு வகையை படவடிவில் உருவமைக்கலாம்.



Let main() contains	
p1:=Person()	statement creates the object.
firstName := " Padmashri "	setting a field called firstName with value Padmashri
lastName :="Baskar"	setting a field called lastName with value Baskar
id := "994-222-1234"	setting a field called id value 994-222-1234
email="compsci@gmail.com"	setting a field called email with value compsci@gmail.com
-- output of firstName : Padmashri	

- இனக்குழு கட்டமைப்பு ஒரு நபரை குறிக்கும் பல பகுதி பொருளுக்கு வடிவம் வரையறுக்கிறது. இந்த வரை யறுப்பு புதிய தரவு வகையை சேர்க்கிறது. இதில் நபர் என்பது வகையாகும். ஒரு முறை வரையறுத்தலின்,
- நாம் புதிய மாறிகளை உருவாக்கலாம். இந்த எடுத்துக்காட்டில் Person என்பது இனக்குழு அல்லது வகையாகும். p1 என்பது ஒரு பொருளாகும். இது போன்று இவ்இனக்குழுவைக் கொண்டு பல பொருள்களை உருவாக்கலாம்.

பாடம் - 3 வரையெல்லை

பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

1. வரையெல்லை என்றால் என்ன?

- வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற் கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும்.

2. மாறிகளுக்கு எதற்காக வரையெல்லை பயன்படுத்தப்பட வேண்டும்? காரணம் கூறுக?

- மாறிகளுக்கு வரையெல்லை பயன்படுத்துவது சிறந்த வழிமுறையாகும், ஏனென்றால் அது ஒரே ஒரு வரையறைக்குள் மாறியை உட்படுத்துகிறது.
- மாறியின் வரையெல்லை என்பது அது குறிமுறையின் எங்கு புலப்படுகிறதோ அல்லது காணப்படுகிறதோ அந்தப் பகுதியாகும்.

3. மேப்பிங் என்றால் என்ன?

- மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல் முறையே மேப்பிங் (Mapping) எனப்படும்.
- = செயல்குறி நிரலாக்க மொழியில் மாறி மற்றும் பொருளை மேப் செய்கிறது.

4. Namespaces சிறுகுறிப்பு வரைக?

- namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம்.

5. private மற்றும் protected அணுகியல்புகளை பைத்தான் எவ்வாறு குறிப்பிடுகிறது.

- பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக் கோட்டும் வழக்கத்தைப் பரிந்துரைக்கிறது.
- இதனால் private மற்றும் protected அணுகியல்பு வரையறுப்புகள் சில பண்புகளைப் பின்பற்றுகின்றன.

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

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

1. Disp():	Entire program	Output of the Program
2. a:=7		7
3. print a		
4. Disp()		

- மேலே உள்ள குறிமுறையை இயக்கும் போது, மாறி a என்பது 7 என்ற மதிப்பை வெளியிடுகிறது.
- ஏனெனில், இது உள்ளமை வரையெல்லையில் வரையறுக்கப்பட்டு, அங்கேயே அச்சிடப்படுகிறது.

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

	Entire program	Output of the Program
1. a:=10		7
2. Disp():		10
3. a:=7		
4. print a		
5. Disp()		
6. print a		

- மேலே உள்ள குறிமுறையில் disp () என்ற செயற்கூறு அழைக்கப்படும் போது, அதனுள் வரையறுக்கப்பட்டிருக்கும் மாறி a, 7 என்ற மதிப்பை வெளியிடும், பின்னர் இது 10 என்ற மதிப்பை வெளியிடும்.
- ஏனெனில் a என்ற மாறி முழுதளாவிய வரையெல்லையில் வரையறுக்கப்பட்டுள்ளது.

3. அடைக்கப்பட்ட வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?

- அனைத்து நிரலாக்க மொழிகளும் பின்னலான செயற்கூறுகளை அமைக்க அனுமதிக்கின்றன.
- ஒரு செயற்கூறின் (வழிமுறை) உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது பின்னலான செயற்கூறு எனப்படும்.
- மற்றொரு செயற்கூறு வரையறையை , தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால் , உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, அடைக்கப்பட்ட வரையெல்லை எனப்படும்.
- நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும் பொழுது அது முதலில் உள்ளமை வரையெல்லையில் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும்.

	Entire program	Output of the Program
1. Disp():		10
2. a:=10		10
3. Disp1():		
4. print a		
5. Disp1()		
6. print a		
7. Disp()		

- 'a' என்ற மாறி Disp () என்ற செயற்கூறினுள் வரையறுக்கப்பட்டிருந்தாலும் Disp1 () என்ற செயற்கூறும் அதைப் பயன்படுத்த முடியும். ஏனெனில் இந்த செயற்கூறு Disp () என்பதன் உறுப்பாகும்.

4. அணுகல் கட்டுப்பாடு எதற்குத் தேவைப்படுகிறது?

- அணுகல் கட்டுப்பாடு என்பது கணினி சூழலில் உள்ள வளங்களை யாரெல்லாம் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில்நுட்பமாகும்.
- இது பாதுகாப்பின் ஒரு அடிப்படைக் கருதாகும். பொருளுக்கான ஆபத்தைக் குறைக்கிறது. அதாவது அணுகல் கட்டுப்பாடு என்பது தரவை அணுகுவதற்கான குறிப்பிடப்பட்ட கட்டுப்பாடாகும்.

5. பின்வரும் போலிக் (Pseudo) குறிமுறையில் மாறிகளின் வரையெல்லையைக் கண்டறிந்து வெளியீட்டை எழுதுக.

```

output
color:= Red
mycolor():
  b:=Blue

```

```

myfavcolor():
    g:=Green
    print color, b, g
myfavcolor()
print color, b
mycolor()
print color

```

```

வெளியீடு
Red Blue Green
Red Blue
Red
மாறியின் வரையெல்லை
color:='Red'   முழுதளாவிய வரையெல்லை
b:='Blue'     அடைக்கப்பட்ட வரையெல்லை
g:='Green'    உள்ளமை வரையெல்லை

```

பகுதி II

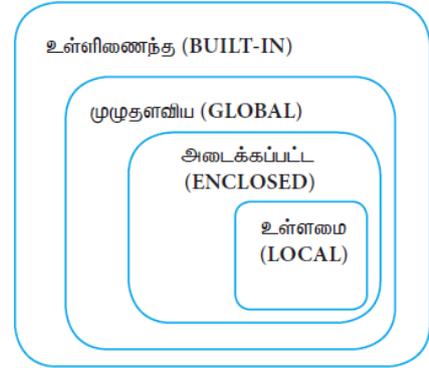
அனைத்து வினாக்களுக்கும் விடையளி

1. மாறியின் வரையெல்லைகளின் வகைகளை விளக்குக (அல்லது) LEGB விதியை எடுத்துக்காட்டுடன் விளக்குக?

LEGB விதி

- LEGB விதி வரையெல்லை தேடப்பட வேண்டிய வரிசையை தீர்மானிக்கப் பயன்படுகிறது.
- வரையெல்லைகள் பின்வருமாறு படிநிலைமுறையில்பட்டியலிடப்பட்டுள்ளன. (பெரியதிலிருந்து சிறியது)

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



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

1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program Disp() a:=7 print a Disp ()	Output of the Program 7
--	--	--------------------------------

- மேலே உள்ள குறிமுறை யை இயக்கும் போது, மாறி a என்பது 7 என்ற மதிப்பை வெளியிடுகிறது.
- ஏனெனில், இது உள்ளமை வரையெல்லையில் வரையறுக்கப்பட்டு, அங்கேயே அச்சிடப்படுகிறது.

முழுதளவிய வரையெல்லை

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

Code	Entire program	Output of the Program
1. a:=10		
2. Disp():		7
3. a:=7		
4. print a		10
5. Disp()		
6. print a		

- மேலே உள்ள குறிமுறை யில் disp () என்ற செயற்கூறு அழைக்கப்படும் போது, அதனுள் வரையறுக்கப்பட்டிருக்கும் மாறி a, 7 என்ற மதிப்பை வெளியிடும், பின்னர் இது 10 என்ற மதிப்பை வெளியிடும்.
- ஏனெனில் a என்ற மாறி முழுதளாவிய வரையெல்லையில் வரையறுக்கப்பட்டுள்ளது.

அடைக்கப்பட்ட வரையெல்லை

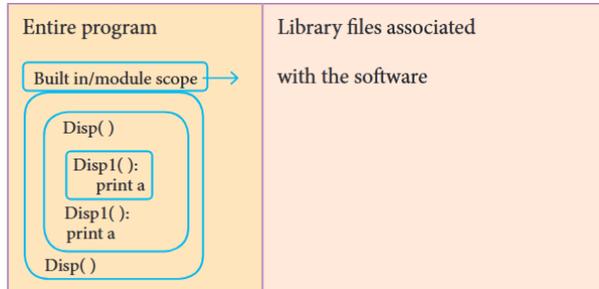
- அனைத்து நிரலாக்க மொழிகளும் பின்னலான செயற்கூறுகளை அமைக்க அனுமதிக்கின்றன.
- ஒரு செயற்கூறின் (வழிமுறை) உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது பின்னலான செயற்கூறு எனப்படும்.
- மற்றொரு செயற்கூறு வரையறையை , தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால் , உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, அடைக்கப்பட்ட வரையெல்லை எனப்படும்.
- நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும் பொழுது அது முதலில் உள்ளமை வரையெல்லையில் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும்.

Code	Entire program	Output of the Program
1. Disp():		
2. a:=10		10
3. Disp():		10
4. print a		
5. Disp()		
6. print a		
7. Disp()		

- 'a' என்ற மாறி Disp () என்ற செயற்கூறினுள் வரையறுக்கப்பட்டிருந்தாலும் Disp1 () என்ற செயற்கூறும் அதைப் பயன்படுத்த முடியும். ஏனெனில் இந்த செயற்கூறு Disp () என்பதன் உறுப்பாகும்.

உள்ளிணைந்த வரையெல்லை

- நிரலாக்க மொழியின் நூலக செயற்கூறினுள் வரையறுக்கப்பட்ட மாறி அல்லது செயற்கூறு உள்ளிணைந்த வரையெல்லையைக் கொண்டுள்ளது.
- இவைகள், நூலக கோப்புகள் நிரலில் செயல்பட தொடங்கியவுடன் இறக்கப்படும்.



2. தொகுதிகளின் ஐந்து பண்பியல்புகளை எழுதுக?

1. தொகுதிகள் தரவு, தகவல் மற்றும் தருக்க செயலாக்கத்தைக் கொண்டுள்ளன.
2. தொகுதிகள் தனியாக தொகுக்கப்பட்டு நூலகத்தில் சேமிக்கப்படும்.
3. தொகுதிகள் நிரலில் சேர்க்க முடியும்.
4. ஒரு பெயரையும், சில அளபுருக்களையும் பயன்படுத்தி தொகுதி பிரிவுகள் செயல்படுத்தப்படுகின்றன.
5. ஒரு தொகுதியின் பிரிவுகள் மற்ற தொகுதிகளால் பயன்படுத்தப்படுகின்றன.

3. தொகுதி நிரலாக்கத்தின் பயன்களை எழுதுக?

- குறைந்த வரிகளைக் கொண்ட குறிமுறையை எழுதினால் போதுமானது.
- மறுபயனாக்கத்திற்கும் பலமுறை குறிமுறை தட்டச்சு செய்வதை தவிர்ப்பதற்கு, ஒற்றை செயல் முறையை உருவாக்க வேண்டும்.
- நிரல்கள் மிக எளிதாக வடிவமைக்கப்படுகின்றன. ஏனெனில், முழு குறிமுறையும் சிறிய பகுதிகளாக பிரிக்கப்பட்டு சிறிய குழுவினரால் கையாளப்படுகிறது.
- தொகுதி நிரலாக்கம் பல நிரலர்களை ஒரே பயன்பாட்டில் வேலை செய்ய அனுமதிக்கிறது.
- பல கோப்புகளில் இந்த குறிமுறை சேமிக்கப்படுகிறது.
- குறிமுறை சிறியதாக, எளியதாக, புரிந்து கொள்ளும் வகையில் உள்ளது.
- துணை நிரல்களாக அல்லது செயல் கூறுகளாக இருப்பதால் பிழைகளை எளிதாக கண்டு பிடிக்க இயலும்.
- ஒரே குறிமுறை பல பயன்பாடுகளில் பயன்படுத்தப்படலாம்.
- மாறிகளின் வரையெல்லையை எளிதில் கட்டுப்படுத்த முடியும்.

பாடம் - 4 நெறிமுறையின் யுக்திகள் பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

1. நெறிமுறை என்றால் என்ன?

- நெறிமுறை என்பது குறிப்பிட்ட செயலை முடிப்பதற்கான வரையறுக்கப்பட்ட கட்டளைகளின் தொகுதியாகும்.
- கொடுக்கப்பட்ட சிக்கலைத் தீர்க்கும் படிநிலை உடைய செய்முறை ஆகும்.

2. முன்மதிப்பீடு மற்றும் பின்சோதனை வேறுபடுத்துக.

முன்மதிப்பீடு	பின்சோதனை
இது நெறிமுறையின் கோட்பாட்டு செயல்திறன் பகுப்பாய்வு ஆகும்.	இது செயல்திறன் அளவிடுதல் ஆகும்.
வெளிப்புற காரணிகளை ஏற்றுக் கொண்டு நெறிமுறையின் செயல்திறன் அளக்கப்படுகிறது.	இந்த பகுப்பாய்வில் உண்மையான புள்ளி விவரங்களான இயங்கும் நேரம் மற்றும் நிரல் நெறிமுறை நிறைவேற்ற தேவைப்படும் இடம் ஆகியவை சேகரிக்கப்படுகிறது.

3. செருகும் வரிசையாக்கம் என்றால் என்ன?

- நெறிமுறையின் முடிவில் இறுதியாக வரிசையாக்கம் செய்யப்பட்ட அணியினை அமைக்கும்.
- இது அணியின் கீழ்ப்பகுதியில் வரிசையாக்கம் செய்யப்பட்ட துணைப் பட்டியலை எப்பொழுதும் அமைத்துக்கொள்ளும்.

4. வரிசையாக்கம் என்றால் என்ன?

- ஒரு குறிப்பிட்ட முறையில் வழிமுறைகளைப் பயன்படுத்தி உருப்படிகளை வரிசைப்படுத்தும் முறையே வரிசையாக்கம் எனப்படும்.
- எடுத்துக்காட்டு:
 - குமிழி வரிசையாக்கம்
 - செருகும் வரிசையாக்கம்
 - தேர்ந்தெடுப்பு வரிசையாக்கம்

5. தேடல் என்றால் என்ன? அதன் வகைகளை எழுதுக.

- தரவு கட்டமைப்பில் உள்ள ஒரு உருப்படியைத் தேடும் முறையே தேடல் எனப்படும்.
- எடுத்துக்காட்டு:
 - வரிசைமுறைத் தேடல்
 - இருமத் தேடல்

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

1. நெறிமுறையின் பண்பியல்புகளைப் பட்டியலிடுக.

- உள்ளீடு
- வெளியீடு

- எல்லையுற்றது
- வரையறுத்தல்
- செயல்தன்மை
- உண்மைத்தன்மை
- எளிமை
- குழப்பமற்றது
- செயலாக்கம்
- அடக்கமானது
- சார்பற்றது

2. சிக்கல்தன்மை மற்றும் வகைகளைப் பற்றி விவாதிக்க.

- நெறிமுறையின் செயல்திறனை இரண்டு முக்கிய காரணிகளான நேரம் மற்றும் இடம் ஆகியவை தீர்மானிக்கிறது.
 - i) நேரசிக்கல்
 - ஒரு நெறிமுறை செயலை செய்து முடிக்க எண்ணிக்கையே நெறிமுறையின் நேரசிக்கல் எனப்படும்.
 - ii) இடசிக்கல்
 - ஒரு நெறிமுறையின் செயல்பாடு முடியும்வரை அதற்கு தேவைப்படும் நினைவக இடமே இடச்சிக்கல் எனப்படும்.
 - நெறிமுறைக்கு தேவைப்படும் இடம் பின்வரும் இரு கூறுகளின் கூட்டுத் தொகையாகும்.
 - நிலையான பகுதி
 - மாறும் பகுதி

3. இடம் மற்றும் இடசிக்கல்களின் மீது தாக்கத்தை ஏற்படுத்தும் காரணிகள் யாவை?

- நெறிமுறையின் செயல்திறன், நேரத்தையும், நினைவக இருப்பிடத்தையும் எவ்வாறு அது திறமையாகப் பயன்படுத்துகிறது என்பதைத் பொறுத்ததாகும்.

காரணிகள்:

- இயந்திரத்தின் வேகம்
- நிரல்பெயர்ப்பி மற்றும் பிற கணினி மென்பொருள் கருவிகள்
- இயக்க அமைப்பு பயன்படுத்தப்பட்ட நிரலாக்க மொழி தரவு தொகுதி

4. Asymptotic குறியீடு குறிப்பு வரைக.

- Asymptotic குறியீடுகள் நேரம் மற்றும் இடச்சிக்கலைப் பற்றிய அர்த்தமுள்ள கூற்றுகளைப் பயன்படுத்தும் ஒரு மொழியாகும்.
- நேரச் சிக்கலைக் குறிக்கும் மூன்று Asymptotic குறியீடுகள்
 - Big O – மோசமான நிலை
 - Big Ω – சிறந்த நிலை
 - Big Θ – மிதமான நிலை

5. இயங்கு நிரலாக்கத்தைப் பற்றி நீவிர் அறிவின யாவை?

- இயங்கு நிரலாக்க மாணது சிக்கலைத் தீர்த்து சிறிய துணை சிக்கல்களாக பிரித்து அதன்பிறகு மீண்டும் அதை பயன்படத்துவதற்கு பயன்படுகிறது.
- இயங்கு நிரலாக்கத்தின் படிநிலைகள் சிக்கல்களை சிறிய ஒன்றோடு ஒன்றிணைந்த துணை சிக்கல்களாகப் பிரிக்க வேண்டும்.
- சிறிய துணை சிக்கல்களின் உகந்த தீர்வைப் பயன்படுத்தி, சிக்கலின் உகந்த தீர்வை அடைய வேண்டும்.
- இயங்கு நிரலாக்கம் நினைவிருத்தலை பயன்படுத்துகிறது.

பகுதி II

அனைத்து வினாக்களுக்கும் விடையளி

1. நெறிமுறையின் பண்பியல்புகளை விவரி.

உள்ளீடு	பூஜ்ஜியம் அல்லது அதிக எண்ணிக்கையில் வழங்கப்பட வேண்டும்.
வெளியீடு	குறைந்தபட்சம் ஒன்றாவது உருவாக்கப்பட வேண்டும்.
எல்லையுற்றது	வரையறுக்கப்பட்ட எண்ணிக்கையிலான படிநிலைகளில் நெறிமுறை நிறுத்தப்பட வேண்டும்.

வரையறுத்தல்	அனைத்து செயல்பாடுகளும் நன்றாக வரையறுக்கப்பட வேண்டும்.
செயல்தன்மை	ஒவ்வொரு கட்டளைகளும் திறம்பட நடத்தப்பட வேண்டும்.
உண்மைத்தன்மை	நெறிமுறைகள் பிழைஇல்லாததாக இருக்கவேண்டும்.
எளிமை	செயல்படுத்துவதற்கு மிக எளிதாக இருக்கவேண்டும்.
குழப்பமற்றது	நெறிமுறையானது தெளிவாகவும் குழப்பமற்றதாகவும் இருத்தல் வேண்டும். படிநிலைகளும் மற்றும் அதனுடைய உள்ளீடுகள்/ வெளியீடுகள் தெளிவானதாக இருக்க வேண்டும் ஒரே ஒரு பொருளுக்கு வழி வகுக்க வேண்டும்.
செயலாக்கம்	கிடைக்கும் வளங்களை வைத்து செயலாக்கவல்லது.
அடக்கமானது	நெறிமுறை பொதுவானதாக இருக்க வேண்டும். அனைத்து வகையான உள்ளீடுகளையும் கையாள்வதற்கு எந்த நிரலாக்க மொழியையும் மற்றும் இயக்க அமைப்பையும் சாராமல் இருக்க வேண்டும்.
சார்பற்றது	நெறிமுறையானது படிநிலை வழிமுறைகளைக் கொண்டிருக்க வேண்டும். அது எந்த நிரலாக்க குறிமுறையை சாராமல் இருக்க வேண்டும்.

2. வரிசைமுறை தேடல் நெறிமுறையை விவாதிக்கவும்.

- வரிசைமுறைத் தேடல் அல்லது தொடர் தேடல் பட்டியலில் விரும்பும் உறுப்பைக் கண்டுபிடிக்கும் வரை அல்லது பட்டியல் முடியும் வரை வரிசை யிலுள்ள ஒவ்வொரு உறுப்புகளையும் சரிபார்த்து, குறிப்பிட்ட மதிப்பைப் பட்டியலில் கண்டுபிடிக்கும் வழிமுறையாகும்.
- பட்டியலை வரிசைப்படுத்த வேண்டிய தேவை இல்லை.

வழிமுறை

1. for மடக்கினைப் பயன்படுத்தி அணியில் பயணித்தல்.
2. ஒவ்வொரு சுழற்சியிலும், இலக்கு மதிப்பை தற்போதைய மதிப்புடன் ஒப்பிடவும்.
 - மதிப்புகள் பொருத்தமாக இருந்தால் அணியின் தற்போதைய சுட்டெண்ணைத் திருப்பி அனுப்பும்.
 - மதிப்புகள் பொருந்தா விட்டால் அணியில் அடுத்துள்ள உறுப்புக்கு சென்று விடும்.
3. பொருத்தம் எதுவும் இல்லையென்றால், -1 மதிப்பைத் திருப்பி அனுப்பும்.

எடுத்துக்காட்டு:

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

- அணியில் 25 என்ற எண்ணைத் தேடுவதற்கு, வரிசைமுறைத் தேடலானது படிப்படியாக தொடர் வரிசையில் தேடலைத் தொடங்கும்.
- இந்த எடுத்துக்காட்டில், எண் 25 சுட்டெண் 3-ல் காணப்படுகிறது.

3. இருமத் தேடல் என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.

- இருமத் தேடலை பாதி இடைவெளித் தேடல் நெறிமுறை என்றும் அழைக்கலாம்.
- வரிசைப்படுத்தப்பட்ட அணிக்குள் இலக்கு மதிப்பின் இருப்பிடத்தைக் கண்டுபிடிக்கிறது.

இருமத் தேடலுக்கான வழிமுறை

1. மைய உறுப்பிலிருந்து தொடங்கவும் :
 - இலக்கு மதிப்பும் அணியின் மைய உறுப்பும் நிகர் எனில் (i.e. 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 மதிப்பைத் திருப்பி அனுப்பும் அல்லது தேடல் நிறைவேற்றப்படவில்லை என்ற தகவலை அறிவிக்கவும்.

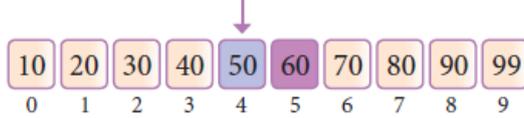
இருமத் தேடல் இயங்கும் கோட்பாடுகள்

- இருமத் தேடலைப் பயன்படுத்தி மதிப்பு 60-ன் இருப்பிடத்தைத் தேடுவதாக எடுத்துக்கொள்வோம்.



அணியின் மைய உறுப்பை தீர்மானிக்கும் வாய்ப்பாடு:

- $mid = low + (high - low) / 2$, $[0 + (9 - 0) / 2 = 4]$ இங்கு அணியின் மையம் 4 ஆகும்.



- 4-ம் சுட்டெண் இருப்பிடத்தில் உள்ள மதிப்பு 50 யை, தேடப்படும் மதிப்பு 60 உடன் ஒப்பிடு செய்வோம். இது தேடப்படும் மதிப்பு கிடையாது.

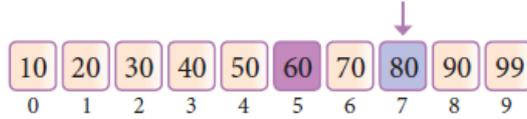


- தேடப்படும் மதிப்பானது 50-விட அதிகமாக இருப்பதால் low மதிப்பை $m + 1$ என மாற்றி புதிய mid மதிப்பை மறுபடியும் கண்டுபிடிக்க வேண்டும்.

$$low = mid + 1$$

$$mid = low + (high - low) / 2 \quad [5 + (9 - 5) / 2 = 5 + 2 = 7]$$

- இப்பொழுது நமது mid மதிப்பு 7 ஆகும்.

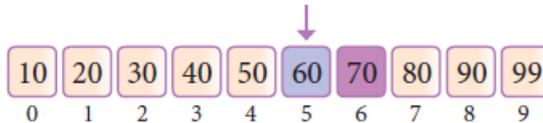
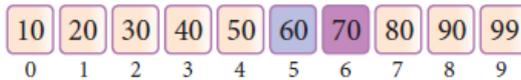


- இருப்பிடம் 7-ல் சேமிக்கப்பட்ட மதிப்பை 60 உடன் ஒப்பிடுவோம். இது தேடப்படும் மதிப்பு கிடையாது. மாறாக, நாம் தேடுவதை விட அதிகமான மதிப்பாக இருப்பதால் high மதிப்பை $m - 1$ என மாற்றி புதிய mid மதிப்பை மறுபடியும் கண்டுபிடிக்க வேண்டும்.

$$high = mid - 1$$

$$mid = low + (high - low) / 2 \quad [5 + (6 - 5) / 2 = 5 + 0.5 = 5]$$

- தற்பொழுது mid மதிப்பு 5 ஆகும்.



- இருப்பிடம் 5-ல் உள்ள சேமிக்கப்பட்ட மதிப்பை இலக்கு மதிப்போடு ஒப்பிடுவோம். இது ஒரு சரியான பொருத்தமாகும்.



- 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 உருப்புகளை ஒப்பீடு செய்து, வரிசையாக்கம் செய்யப்படாத வரிசை எணில் அவற்றை இடமாற்றம் செய்யும்.

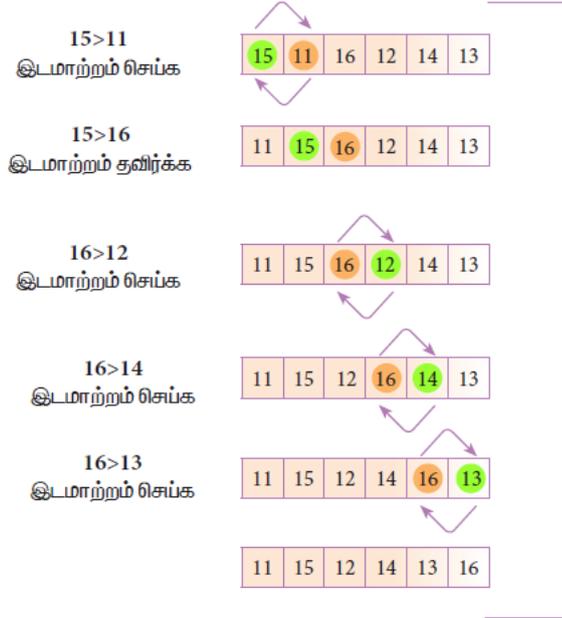
- இடமாற்றம் தேவைப்படும் வரை அவை மீண்டும் மீண்டும் பட்டியலிடப்படும். இது மிகவும் மெதுவானது.

வழிமுறை

- முதல் உறுப்புடன் (சுட்டெண் = 0), அணியின் தற்போதைய உறுப்போடு அடுத்த உறுப்பை ஒப்பீடு செய்யவும்.
- தற்போதைய உறுப்பு அடுத்த உறுப்பை விட அதிகம் எனில், அவற்றை இடமாற்றம் செய்யவும்.
- தற்போதைய உறுப்பு அடுத்த உறுப்பை விட சிறியது எனில், அடுத்த உறுப்பிற்கு செல்லவும் மீண்டும் படிநிலை -1லிருந்து தொடங்கவும்.

எடுத்துக்காட்டு

- {15, 11, 16, 12, 14, 13} மதிப்புகளோடு கூடிய அணியை எடுத்துக்கொள்வோம்.



- மேலே குறிப்பிடப்பட்டுள்ளதை முதல் சுழற்சி படமாகும். இதேபோல், எல்லா சுழற்சி செய்யப்படும்.

- இறுதி சுழற்சிக்கு பிறகு வரிசையாக்கம் செய்யப்பட்ட அணியை கொடுக்கும்.

5. இயங்கு நிரலாக்கத்தின் கருத்துருவை பொருத்தமான எடுத்துக்காட்டுடன் விளக்கவும்.

- இயங்கு நிரலாக்கம் என்பது சிக்கல்களை ஒரே மாதிரியான துணை சிக்கல்களாக பிரித்து, அதன் மூலம் கிடைக்கும் தீர்வை மீண்டும் பயன்படுத்துகிறது.
- பெரும்பாலும், இந்த நெறிமுறைகள் உகந்த தீர்வாகப் பயன்படுத்தப்படுகிறது.
- கையில் உள்ள துணை சிக்கல் தீர்ப்பதற்கு முன் இந்த செயல்முறையானது ஏற்கனவே தீர்வு காணப்பட்ட துணை சிக்கல்களின் முடிவுகளை ஆராய முயற்சிக்கும்.
- மிகச் சிறந்த தீர்வை அடைவதற்கு துணை சிக்கல்களின் தீர்வுகளை ஒன்றிணைத்தல் வேண்டும்.

இயங்கு நிரலாக்கத்தின் படிநிலைகள்

- சிக்கல்களை சிறிய ஒன்றோடு ஒன்றிணைந்த துணை சிக்கல்களாகப் பிரிக்க வேண்டும்.
- சிறிய துணை சிக்கல்களின் உகந்த தீர்வைப் பயன்படுத்தி, சிக்கலின் உகந்த தீர்வை அடைய வேண்டும்.
- இயங்கு நிரலாக்கம் நினைவிருத்தலை பயன்படுத்துகிறது.

எடுத்துக்காட்டு

- பைபோனாசி வரிசையானது முந்தைய இரண்டு எண்களை கூட்டி அடுத்தடுத்த எண்களை உருவாக்கும்.
- பின்வரும் நிபந்தனைகளை பைபோனாசி வரிசை நிறைவேற்றும்:

$$Fib_n = Fib_{n-1} + Fib_{n-2}$$

பைபோனாசி சுழற்சி நெறிமுறை

$f_0=0, f_1=1$ என தொடக்க மதிப்பிருத்தல் வேண்டும்

படிநிலை -1: Print the initial values of Fibonacci f0 and f1

படிநிலை -2: fib ←f0 + f1 என மதிப்பிருத்தல் வேண்டும்

படிநிலை -3: மதிப்பிருத்தல் f0←f1, f1←fib

படிநிலை -4: பைபோனாசியின் அடுத்த மதிப்பை fib காண்பிக்கவும்

படிநிலை -5: குறிப்பிட்ட வரிசை உருவாகும் படிநிலை-2 வரை திரும்பச் செய்தல்

- உள்ளீடு n = 10
- 10 இலக்க வரை பைபோனாசி நெறிமுறை கீழே கொடுக்கப்பட்டுள்ளது:
- பைபோனாசி வரிசை : 0 1 1 2 3 5 8 13 21 34 55

பாடம்-5: பைத்தான் அறிமுகம் - மாறிகள் மற்றும் செயற்குறிகள்
பகுதி-ஆ

பின்வரும் வினாக்களுக்கு விடையளி: (2 மதிப்பெண்)

1. பைத்தான் நிரலினை சோதிக்க எத்தனை வகை முறைமைகள் உள்ளன?

- பைத்தான் நிரலினை சோதிக்க இரண்டு வகையான முறைமைகள் உள்ளன. அவை
 - ஊடாடும் முறைமை
 - ஸ்கிரிப்ட் முறைமை

2. வில்லைகள் பற்றி சிறு குறிப்பு வரைக.

- பைத்தான், நிரலில் இடம்பெறும் நிரல் வரிகளை அடிப்படை மொழிக் கூற்றுகளாக பிரிக்கிறது. இந்த கூறுகள் வில்லைகள் எனப்படும்.
- வில்லைகளின் வகைகள்
 - குறிப்பெயர்கள்
 - சிறப்புச்சொற்கள்
 - செயற் குறிகள்
 - வரம்புக்குறிகள்
 - நிலைஉருக்கள்

3. பைத்தானில் உள்ள பல்வேறு செயற்குறிகள் யாவை ?

- கணித செயற்குறிகள்
- உறவுநிலை அல்லது ஒப்பீடு செயற்குறிகள்
- தருக்க செயற்குறிகள்
- மதிப்பிருத்து செயற்குறிகள்
- நிபந்தனை செயற்குறி

4. குறிப்பெயர்கள் என்றால் என்ன ? குறிப்பெயர்கள் வகைகள் யாவை ?

- மாறி, செயற்கூறு, இனக்குழு, தொகுதி அல்லது பொருள்களை அடையாளம் காண்பயன்படும் பெயர்கள் குறிப்பெயர் என்று அழைக்கப்படுகிறது.

குறிப்பெயர்கள் வகைகள்

- எழுத்துக்கள் (அ) அடக்கீறு (_) கொண்டு தொடங்கலாம்
- எண்கள் கொண்டிருக்கலாம்
- எழுத்து வடிவுணர்வு கொண்டது
- பைத்தான் சிறப்புச் சொற்களாக இருக்க கூடாது
- %, \$, @ போன்ற நிறுத்தற்குறிகள் இடம் பெறக்கூடாது

5. அடுக்கெண் தரவு பற்றி குறிப்பு வரைக.

- அடுக்கெண் தரவு வகை , தசம எண் பகுதி, தசம புள்ளி, அடுக்கெண் பகுதியை தொடர்ந்து ஒன்று அல்லது பல இலக்கங்களை கொண்டிருக்கும்.
- எடுத்துக்காட்டு: 12.E04, 24.e04

பகுதி-இ

பின்வரும் வினாக்களுக்கு விடையளி: (3 மதிப்பெண்)

1. கணித செயற்குறிகள் பற்றி குறிப்பு வரைக. எடுத்துக்காட்டு தருக.

- கணித செயற்குறிகள், இரு செயலேற்பிகளை ஏற்றுக் கொண்டு அதன் மீது கணித செயல்பாடுகளை செய்யும்.
- கணித செயற்குறிகள் : +, -, *, /, % (வகுமீதி), ** (அடுக்கு) மற்றும் // (முழு எண் வகுத்தி)

- எடுத்துக்காட்டு:
 - o a=100, b=10 என மதிப்பாக கொண்டால்
 - o (i) a + b = 110 (ii) a % 30 = 10 (iii) a // 30 = 3 (iv) a**2 = 10000

2. பைத்தானில் மதிப்பிருத்தல் செயற்குறிகள் என்றால் என்ன ?

- பைத்தானில், = என்பது ஒரு மதிப்பிருத்தல் செயற்குறியாகும். இது மாறிகளுக்கு மதிப்பிருத்த பயன்படுகிறது.
- பைத்தான், +=, -=, *=, /=, %=, **= மற்றும் //= போன்ற கூட்டு செயற்குறிகளையும் ஏற்கும்.
- எடுத்துக்காட்டு:

```
x=10
x+=20 # x=x+20
x-=5 # x=x-5
x*=5 # x=x*5
```

3. மும்ம செயற்குறியை எடுத்துக்காட்டுடன் எழுதுக.

- மும்ம செயற்குறி, நிபந்தனை செயற்குறி என்றும் அழைக்கப்படுகிறது.
- இது சமன்பாடுகளின் நிபந்தனையை சரி அல்லது தவறா என்று சோதித்து செயல்படுத்தும்.
- மும்ம செயற்குறி பலவரி if..else கூற்று போல் அல்லாது நிபந்தனைகளை ஒற்றை வரியில் சோதிக்க அனுமதிக்கிறது.
- தொடரியல்: Variable Name = [on_true] if [Test expression] else [on_false]
- எடுத்துக்காட்டு: min= 49 if 49<50 else 50

4. விடுபடு வரிசைப்பற்றி குறிப்பு எழுதி எடுத்துக்காட்டு தருக.

- பைத்தானில், பின்சாய்வுகோடு ("\") சிறப்பு குறியீட்டை "விடுபடு" குறியுரு என்று அழைக்கிறோம்.
- இது சில வெற்று இடைவெளிகளை குறிப்பிட பயன்படுகிறது.

விடுபடு குறியுரு	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	குத்தல்	print("Python","\t","Lang..")	Python Lang..

5. சர நிலையுரு என்றால் என்ன ?

- பைத்தானில், சரநிலையுருக்கள் என்பது, மேற்கோள் குறிக்குள் அடைக்கப்பட்ட குறியுருக்களின் தொடராகும்.
- பைத்தானில், சரங்களை ஒற்றை, இரட்டை மற்றும் மூன்று மேற்கோள் குறிகளில் அடைக்கலாம்.
- குறியுரு மதிப்புருக்கள் ஒற்றை அல்லது இரட்டைமேற்கோள்குறியை கொண்டிருக்கும். பலவரி சரநிலையுருவை மூன்று மேற்கோள் "" "" குறிக்குள் கொடுக்கப்பட வேண்டும்.
- எடுத்துக்காட்டு:

```
strings = "This is Python"
char = "C"
multiline_str = """This is a multiline string with more than one line code."""
```

பகுதி-ஈ

பின்வரும் வினாக்களுக்கு விடையளி:

(5 மதிப்பெண்)

1. ஸ்கிரிப்ட் முறைமை நிரலாக்கம் பற்றி எழுதுக.

- அடிப்படையாக, ஒரு ஸ்கிரிப்ட் என்பது பைத்தான் கட்டளைகளை கொண்ட ஒரு உரை ஆவணத்தைக் குறிக்கிறது.
- பைத்தான் ஸ்கிரிப்ட் குறிமுறையை மறுபயனாக்கம் செய்துக் கொள்ளலாம்.

- ஒரு முறை ஸ்கிரிப்ட் கோப்பை எழுதிவிட்டால் பல முறை இயக்கிக் கொள்ளலாம் மீண்டும் அந்த ஸ்கிரிப்டை உருவாக்க தேவையில்லை . ஸ்கிரிப்ட் பதிப்பாய்வு செய்ய கூடியவை.

பைத்தான் ஸ்கிரிப்ட் எழுதுதல்

- பைத்தான் shell சாளரத்தில் File → New File அல்லது Ctrl + N என்பதை அழுத்தவும்.
- Untitled வெற்று ஸ்கிரிப்ட் Text editor தோன்றும்.
- பின்வரும் கட்டளைகளை ஸ்கிரிப்ட் editorல் உள்ளிடவும்.

```
a =100
b = 350
c = a+b
print ("The Sum=", c)
```

பைத்தான் ஸ்கிரிப்ட்டை சேமித்தல்

- File → Save கட்டளையை தேர்ந்தெடுக்கவும் அல்லது Ctrl + S அழுத்தவும்.
- இப்பொழுது, Save As உரையாடல் பெட்டி திரை தோன்றும்.
- Save As உரையாடல் பெட்டியில், கோப்பு சேமிக்கப்பட வேண்டிய இடத்தை தேர்ந்தெடுக்க வேண்டும்.
- பின்னர் கோப்பின் பெயரை File Name பெட்டியில் உள்ளிட வேண்டும். கொடாநிலையாக பைத்தான்
- கோப்புகள் .py என்ற நீட்டிப்புடன் சேமிக்கப்படுகிறது.
- இறுதியாக Save பொத்தானை தேர்ந்தெடுத்து, பைத்தான் ஸ்கிரிப்ட்டை கோப்பை சேமிக்கலாம்.

பைத்தான் ஸ்கிரிப்ட்டை இயக்குதல்

- Run → Run Module அல்லது F5 யை தேர்ந்தெடுக்கவும்.
- குறிமுறையில் பிழைகள் இருப்பின் அவை சிகப்பு நிறத்தில் IDLE திரையில் காண்பிக்கப்பட்டு, பைத்தான் பிழைக்கான காரணத்தை விளக்கும்.
- பிழைகளை திருத்துவதற்கு, ஸ்கிரிப்ட் editorக்கு சென்று, பிழை களைத் திருத்தி, கோப்பை Ctrl + S அல்லது File → Save கட்டளைக் கொண்டு சேமித்தபின் மீண்டும் இயக்க வேண்டும்.
- பிழைகள் இல்லாத பைத்தான் குறிமுறையை இயக்கியவுடன் அதன் வெளியீடு பைத்தான் IDLE திரையில் தோன்றும்.

2. input() மற்றும் print() செயற் கூறுகள் பற்றி விளக்கு.

print() செயற்கூறு

- பைத்தானில், print() செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை வெளியிட பயன்படுகிறது.
- print() செயற்கூறின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது

```
print ("string to be displayed as output " )
print (variable )
print ("String to be displayed as output ", variable)
print ("String1 ", variable, "String 2", variable, "String 3" .....
```
- எடுத்துக்காட்டு:

```
print ("Welcome to Python Programming")
print (z)
print ("The sum = ", z)
print ("The sum of ", x, " and ", y, " is ", z)
```

input()செயற்கூறு

- பைத்தானில், input() செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடாக பெற்றுக் கொள்ளப்பயன்படுகிறது.
- input() செயற்கூறின் தொடரியல், Variable = input ("prompt string")
- இந்த தொடரியலில் உள்ள "prompt string" பயனர் கொடுக்கப்பட வேண்டிய உள்ளீடு எதுவென்பதை உணர்த்தும் கூற்று அல்லது செய்தியை கொண்டிருக்கும்.
- எடுத்துக்காட்டு 1: city=input ("Enter Your City: ")
- எடுத்துக்காட்டு 2: city=input()
- மேலே உள்ள எடுத்துக்காட்டு 2ல், input () செயற் கூற்றில் "prompt string" இடம்பெறவில்லை. எனவே, எவற்றை உள்ளீடாக தர வேண்டும் என்பது பயனர் புரிந்து கொள்ள இயலாது.

- input() செயற்கூறு அனைத்து தரவுகளையும் சரங்களாக ஏற்றுக் கொள்ளும். பயனர் எண்களை மதிப்புகளாக உள்ளீடு செய்யும் போது, அவை வெளிப்படையாக எண்வகை தரவுகளாக மாற்றப்பட வேண்டும். int() செயற்கூறு சர வகை தரவை எண்வகை தரவாக வெளிப்படையாக மாற்ற உதவுகிறது.
- எடுத்துக்காட்டு 3: x = int (input("Enter Number 1: "))

3. பைத்தானில் உள்ள வில்லைகள் பற்றி எழுதுக.

- பைத்தான், நிரலில் இடம்பெறும் வரிகளை அடிப்படை சொற்களாக பிரிக்கிறது. இந்த கூறுகள் வில்லைகள் எனப்படும்.

i. குறிப்பெயர்கள்:

- மாறி, செயற்குறி, இனக்குழு, தொகுதி மற்றும் பொருளின் பெயர்களை குறிப்பெயர்கள் என்கிறோம்.

குறிப்பெயர்கள் வகைகள்:

- எழுத்துக்கள் (அ) அடிக்கீறு (_) கொண்டு தொடங்கலாம்
- எண்கள் கொண்டிருக்கலாம்
- எழுத்து வடிவணர்வு கொண்டது
- பைத்தான் சிறப்புச் சொற்களாக இருக்க கூடாது
- %, \$, @ போன்ற நிறுத்தற்குறிகள் இடம் பெறக்கூடாது
- எ.கா: Sum, total_marks, num1

ii. சிறப்புச் சொற்கள்:

- நிரலின் அமைப்பை அடையாளம் காண பைத்தான் சிறப்புச் சொற்களை பயன்படுத்துகிறது
- மொழி பெயர்ப்பியில் குறிப்பிட்ட பொருள் கொண்டுள்ளதால் இவற்றை பிற பயன்பாட்டிற்கு பயன்படுத்தக்கூடாது.
- எ.கா: for, return, while

iii. செயற்குறிகள்:

- கணிப்பீடுகள், நிபந்தனை சோதிப்பு போன்ற செயல்பாடுகளை செய்ய பயன்படுகிறது.
- செயற்குறிகள் செயலேற்பிகளில் உள்ள மதிப்புகளில் செயல்படும்
- வகைகள்:
 - கணித செயற்குறிகள் (+, -, *, /, %, //)
 - தொடர்புடைய (அ) ஒப்பீடு செயற்குறிகள் (<, >, <=, >=, ==, !=)
 - தருக்க செயற்குறிகள் (and, or, not)
 - மதிப்பிருத்து செயற்குறிகள் (=, +=, -=, *=, /=, %=)
 - நிபந்தனை செயற்குறி

iv. வரம்புக்குறி

- பைத்தான், குறியீடு (அ) குறியீடுகளின் தொகுப்பை, கோபை, பட்டியல், அகராதி மற்றும் சரங்களில் பயன்படுத்துகிறது.
- எ.கா: (), [], { }, ' ,

v. நிலைஉரு

- மாறிகள் (அ) மாறிலிகளுக்கு வழங்கப்படும் மூல தரவு
- வகைகள்:
 - எண்கள்
 - சரம்
 - பூலியன்

பாடம்-6: கட்டுப்பாட்டு கட்டமைப்புகள் பகுதி-ஆ

பின்வரும் வினாக்களுக்கு விடை யளி:

(2 மதிப்பெண்)

1. பைத்தானில் உள்ள கட்டுப்பாட்டு கட்டமைப்புகளைப் பட்டியலிடுக.

- வரிசைமுறை கூற்றுகள்
- மாற்று அல்லது கிளைப்பிரிப்பு
- பன்முறைசெயல் அல்லது மடக்கு

2. break கூற்றைப் பற்றி குறிப்பு வரைக.

- மடக்கை விட்டு வெளியேறச் செய்கிறது

- மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது.
- பின்னலான மடக்கின் உள்ளே இருந்தால் break கூற்று மிக உட்புறமாக அமைந்துள்ள மடக்கை விட்டு வெளியேறும்.
- தொடரியல்: break

3. if..else கூற்றின் பொது வடிவத்தை எழுதுக?

```
if <condition>:
    statements-block 1
else:
    statements-block 2
```

4. கட்டுப்பாட்டு கட்டமைப்பு என்றால் என்ன ?

- கட்டுப்பாடு நிரலின் ஒரு பகுதியில் இருந்து இன்னொரு பகுதிக்கு தாவுவதற்கு காரணமான நிரல் கூற்றுகள் கட்டுப்பாட்டு கட்டமைப்பு அல்லது கட்டுப்பாட்டு கூற்றுகள் எனப்படும்.

5. range() செயற்கூறு குறிப்பு வரைக.

- range() செயற்கூறு ஒரு உள்ளிணைந்த செயற்கூறாகும். இது, இரண்டு எண் இடைவெளிக்குட்பட்ட மதிப்புகளை உருவாக்கும்.
- range()ன் தொடரியல்: range(start,stop,[step])
- இதில்,
 - start – தொடக்க மதிப்பைக் குறிக்கும்
 - stop – இறுதி மதிப்பைக் குறிக்கும்
 - step – மிகுப்பு மதிப்பைக் குறிக்கும். இது விருப்பப் பகுதியாகும்.
- எடுத்துக்காட்டு


```
range (1,30,1) range-ன் மதிப்புகள் 1 முதல் தொடங்கி 29 வரை
```

பகுதி-இ

பின்வரும் வினாக்களுக்கு விடையளி:

(3 மதிப்பெண்)

1. பின்வரும் வெளியீட்டைப் பெற நிரலை எழுதுக

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

```
str="ABCDE"
i=1
while(i<=len(str)):
    print(str[0:i], sep=" ")
    i+=1
```

2. if..else அமைப்பைப் பற்றி குறிப்பு வரைக.

- if .. else கூற்று சரி தொகுதி மற்றும் தவறு தொகுதி இரண்டையுமே சரிபார்ப்பதற்கான கட்டுப்பாட்டை வழங்குகிறது.
- if..else கூற்று இரு மாற்று வழிகளை வழங்குகிறது. நிபந்தனையானது எந்த தொகுதியை நிறைவேற்றப்பட வேண்டும் என்பதைத் தீர்மானிக்கிறது.
- தொடரியல்:

```
if <condition>:
    statements-block 1
else:
    statements-block 2
```

- எடுத்துக்காட்டு


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

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

```
a=int(input("Enter the first number:"))
```

```

b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))
if (a>b) and (a>c):
    print(a, " is the largest number")
elif (b>c):
    print(b, " is the largest number ")
else:
    print(c, " is the largest number ")

```

4. while மடக்கின் பொதுவடிவம் யாது?

```

while <condition>:
    statements block 1
[else:
    statements block 2]

```

5. break மற்றும் continue கூற்றுகளின் வேறுபாடு யாது?

Break	continue
மடக்கை விட்டு வெளியேறச் செய்கிறது	மடக்கின் அடுத்த சுழற்சிக்கு கொண்டு செல்கிறது
நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது	மடக்கின் மீதமுள்ள குறிமுறையைத் தவிர்த்து அடுத்த மடக்கு செயலை ஆரம்பிக்கும்

பகுதி-ஈ

பின்வரும் வினாக்களுக்கு விடையளி:

(5 மதிப்பெண்)

1. for மடக்கை பற்றி விரிவான விடையளிக்கவும்.

- பொதுவாக, for மடக்கு ஒரு, "வரையறுக்கப்பட்ட மடக்கு" என்று அழைக்கப்படுகிறது.
- ஏனெனில், இந்த மடக்கு சரியாக எத்தனை முறை மீண்டும் மீண்டும் இயங்கும் என்பதை நிரலர் தெரிந்துவைத் திருப்பார்.
- தொடரியல்:


```

for counter_variable in sequence:
    statements-block 1
[else:
    statements-block 2]

```
- for .. in தொடர் பொருள்களின் வழியே சுழலும் தன்மையுடையது. அதாவது, ஒரு தொடரின் ஒவ்வொரு மதிப்பின் ஊடாக செல்லும்.
- இங்கே தொடர் என்பது வரிசைப்படுத்தப்பட்ட அல்லது வரிசைப்படுத்தப்படாத மதிப்புகள் அல்லது ஒரு சரமாகவும் இருக்கலாம்.
- கட்டுப்பாட்டு மாறி, ஒவ்வொரு சுழற்சியிலும், தொடரின் கடைசி மதிப்பு வரை, தொடரின் ஒவ்வொரு மதிப்பையும் அணுகும்.
- எடுத்துக்காட்டு:


```

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

```
- வெயிளிடு: 1 2 3 4 5
- range() செயற்கூறு ஒரு உள்ளிணைந்த செயற்கூறாகும். இது, இரண்டு எண் இடைவெளிக்குட்பட்ட மதிப்புகளை உருவாக்கும்.
- range()ன் தொடரியல்:


```

range(start,stop,[step])

```
- இதில்,
 - start – தொடக்க மதிப்பு
 - stop – இறுதி மதிப்பு
 - step – மிகுப்பு மதிப்பு
- எடுத்துக்காட்டு:


```

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

```
- வெயிளிடு: 2 4 6 8

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

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

- தொடரியல்:

```
if <condition-1>:
    statements-block 1
elif <condition-2>:
    statements-block 2
else:
    statements-block n
```

- மேற்கண்ட if..elif..else பொதுவடிவத்தில், condition-1 சோதிக்கப்படும். condition-1 சரி எனில் statements-block 1 நிறைவேற்றப்படும்.
- இல்லையெனில் கட்டுப்பாடு, condition-2 யைச் சோதிக்கும். condition-2 சரி எனில் statements-block-2 நிறைவேற்றப்படும்.
- இல்லையெனில் else பகுதியில் உள்ள statements-block n நிறைவேற்றப்படும்.
- எடுத்துக்காட்டு:

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))
if (a>b) and (a>c):
    print(a, " is the largest number")
elif (b>c):
    print(b, " is the largest number ")
else:
    print(c, " is the largest number ")
```

3. அனைத்து மூன்று இலக்க ஒற்றைப்படடை எண்களை வெளியிடுவதற்கான நிரலை எழுதுக.

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

4. கொடுக்கப்பட்ட எண்ணின் பெருக்கல் வாய்ப்பாட்டை வெளியிடும் நிரலை எழுதுக.

```
n = int(input("Enter a Number :"))
x = int(input("Enter number of terms :"))
for i in range(1, x+1):
    print(i, " x ", n, " = ", i*n)
```

பாடம் - 7 பைத்தான் செயற்கூறுகள்

பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண்)

1. செயற்கூறு என்றால் என்ன ?

- ஒரு குறிப்பிட்ட செயலினை செய்வதற்காக வடிவமைக்கப்பட்டு பெயரிடப்பட்ட குறிமுறையின் தொகுதியே செயற்கூறு எனப்படும்.
- செயற்கூறு தொகுதி "def " என்ற சிறப்புச் சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் () அடைப்புக்குறியுடன் முடிய வேண்டும்.

2. செயற்கூறின் வகைகளை எழுதுக.

- பயனர் வரையறைத்த செயற்கூறு
- உள்ளிணைந்த செயற்கூறு
- லாம்ப்டா செயற்கூறு
- தற்சுழற்சி செயற்கூறு

3. செயற்கூறுவின் முக்கிய நன்மைகள் யாவை ?

- செயற்கூறுகள் ஒரு நிரலை சிறு தொகுதியாக பிரிக்க உதவுகிறது.
- குறிமுறையின் மறுபயனாக்கத்திற்கு உதவுகிறது.
- செயற்கூற்றின் செயல்பாடுகளை மாற்றம் செய்வது எளிதாகிறது.

4. மாறியின் வரையெல்லை என்றால் என்ன ? அதன் வகைகளைக் குறிப்பிடுக.

- மாறியின் வரையெல்லை என்பது நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும்.
- வரையெல்லை நடப்பு மாறித் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும்.
- வகைகள்
 - உள்ளமை வரையெல்லை (Local)
 - முழுதளவிய வரையெல்லை (Global)

5. முழுதளவிய வரையெல்லை வரையறு

- அனைத்து செயற்கூறுகளுக்கும் மேலாக அறிவிக்கப்படும் மாறி முழுதளவிய மாறி எனப்படும்.
- முழுதளவிய வரையெல்லை உடைய மாறியை நிரலின் எந்த பகுதியிலும் அணுகமுடியும்.
- முழுதளவிய வரையெல்லை கொண்ட மாறியை ஒரு செயற்கூறினுள் அணுகுவதற்கு 'global' சிறப்புச் சொல்லானது பயன்படுகிறது.

6. தன்னைத் தானே அழைக்கும் செயற்கூறில் அடிப்படை நிபந்தனை என்றால் என்ன ?

- தற்சுழற்சி செயற்கூறு தன்னைத்தானே அழைக்கும்.
- ஒரு நிபந்தனையில் நிறுத்தப்படாவிட்டால், ஒரு செயல்முறை காலவரையின்றி செயல்படும்.
- இந்த செயற்பாடு முடிவிலா சுழற்சி எனப்படும்.
- தற்சுழற்சி செயற்கூறில் கொடுக்கப்படும் நிபந்தனை அடிப்படை நிபந்தனை எனப்படும்.

7. தன்னைத்தானே அழைக்கும் செயற்கூறுக்கு வரம்பை எவ்வாறு அமைக்க வேண்டும்? எ.கா தருக.

- தானமைவாக 1000 அழைப்புகளுக்குப் பிறகு பைத்தான் தன்னைத்தானே அழைப்பதை நிறுத்திவிடும்.
- வரம்புகளை sys.setrecursionlimit(limit_value) பயன்படுத்தி மாற்றியமைக்க அனுமதிக்கிறது.
- எடுத்துக்காட்டு

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact(n-1)
print(fact(2000))
```

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண்)

1. உள்ளமை மாறிகளுக்கான விதிமுறைகளை எழுதுக.

- உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.
- செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.
- செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உயிர்பெறும்.
- முறையான அளபுருக்கள் அதன் செயற்கூறுக்கு உள்ளமையானது ஆகும்.

2. பைத்தானிலுள்ள global சிறப்புச்சொல்லுக்கான அடிப்படை விதிமுறைகளை எழுதுக.

- செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக முழுதளவிய வரையெல்லை ஆகும். 'global' என்ற சிறப்புச் சொல்லை பயன்படுத்த வேண்டியதில்லை.
- செயற்கூறினுள் முழுதளவிய மாறியின் மதிப்பை மாற்றியமைக்க 'global' என்ற சிறப்புச்சொல் பயன்படுத்த வேண்டும்.
- செயற்கூறுவிற்கு வெளியே 'global' என்ற சிறப்புச் சொல் எந்த விளைவையும் ஏற்படுத்தாது.

3. செயற்கூறினுள் முழுதளவிய மாறியை மாற்றம் செய்தால் என்ன நிகழும்?

- செயற்கூறின் உள்ளே குளோபல் மாறியை மாற்றியமைக்க முயற்சிக்கும் போது " Unbound Local Error " என்ற பிழை ஏற்படும்.
- ' global ' என்ற சிறப்பு சொல்லைப் பயன்படுத்தாமல் முழுதளவிய மாறியை செயற்கூறின் உள்ளே மாற்றம் செய்ய முடியாது. ஆனால் அணுக மட்டுமே முடியும்.

4. ceil() மற்றும் floor() செயற்கூறுகளை வேறுபடுத்துக.

ceil()	floor()
math.ceil (x)	math.floor (x)
x-யை விட பெரிய அல்லது xக்கு நிகரான சிறிய முழு எண்ணைத் திருப்பி அனுப்பும்.	x-யை விடக்குறைவான அல்லது x-க்கு நிகரான பெரிய முழு எண்ணைத் திருப்பி அனுப்பும்.
import math x= 26.7 print (math.ceil (x)) வெளியீடு: 27	import math x=26.7 print (math.floor (x)) வெளியீடு: 26

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

```
y=int (input("Enter the year:"))
if y % 4== 0:
    print("LEAP YEAR")
else:
    print("NOT LEAP YEAR")
```

6. செயற்கூறில் தொகுப்பு என்பது என்ன ?

- செயற்கூறு திருப்பி அனுப்பும் மதிப்பு மற்றொரு செயற்கூறிற்கு செயலுருபாக, பின்னலான அமைப்பில் பயன்படுத்தினால் அதற்கு தொகுப்பு (composition) என்று பெயர்.
- எடுத்துக்காட்டாக, input() செயற்கூறு மூலம் பயனரிடமிருந்து சரத்தை உள்ளீடாகப் பெற்று eval() செயற்கூறு மூலம் அதன் மதிப்பை மதிப்பீடு செய்ய வேண்டும்.

```
>>> n2 = eval (input ("Enter an arithmetic expression: "))
Enter an arithmetic expression : 12.0+13.0 * 2
>>> n2
38.0
```

7. தற்சுழற்சி செயற்கூறு எவ்வாறு செயல்படுகிறது?

- தற்சுழற்சி செயற்கூறு வெளிப்புற குறிமுறையிலிருந்து அழைக்கப்படும்.
- அடிப்படை நிபந்தனை நிறைவேற்றப்பட்டால் நிரலானது சரியான வெளியீடு கொடுத்து வெளியேறும்.
- இல்லையெனில், செயற்கூறு தேவையான செயற்பாட்டை இயக்கும் மேலும் தற்சுழற்சி முறையில் தன்னைத் தானே அழைத்துக்கொள்ளும்.

8. செயற்கூறினை வரையறுக்கும் போது குறிப்பிடப்பட வேண்டிய குறிப்புகள் யாவை ?

- செயற்கூறு தொகுதி def என்ற சிறப்புச் சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் () அடைப்புக்குறியுடன் முடிய வேண்டும்.
- ஏதேனும் உள்ளீட்டு அளபுருக்கள் இருப்பின் அவற்றை செயற்கூற்றை வரையறுக்கும்போதே () என்ற அடைப்புக்குறிக்குள் கொடுக்க வேண்டும்.
- குறிமுறை தொகுதியானது எப்பொழுதும் முக்காற்புள்ளிக்கு பிறகு உள்தள்ளி வர வேண்டும்.
- "return [கோவை]" கூற்று செயற்கூறுவை முடித்து வைக்கும். விருப்பப்பட்டால் கோவையின் மதிப்பை, அழைக்கும் கூற்றுக்கு திருப்பி அனுப்பும். செயலுருபுகள் இல்லாத return, return None-க்கு நிகரானது.

பகுதி-II

அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண்)

1. செயற்கூறின் வகைகளை எடுத்துக்காட்டுடன் விவரி

- செயற்கூறுகள் என்பது குறிப்பிட்ட செயலை செய்வதற்கான தொடர்புடைய கூற்றுகளின் தொகுதி ஆகும்.
- செயற்கூறுகளின் வகைகள்
 - பயனர் வரையறுக்கும் செயற்கூறுகள்
 - உள்ளிணைந்த செயற்கூறுகள்
 - லாம்ப்டா செயற்கூறுகள்
 - தற்சுழற்சி செயற்கூறுகள்

பயனர் வரையறுக்கும் செயற்கூறுகள்

- பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள் பயனர் வரையறுக்கும் செயற்கூறுகள் எனப்படும்.
- தொடரியல்:

```
def <function_name ([parameter1, parameter2...]) > :  
    <Block of Statements>  
    return <expression / None>
```
- எடுத்துக்காட்டு:

```
def hello():  
    print ("hello – Python")  
    return
```

உள்ளிணைந்த செயற்கூறுகள்

- பைத்தானில் உள்ளடக்கப்பட்ட செயற்கூறுகள் உள்ளிணைந்த செயற்கூறுகள் எனப்படும்.
- எடுத்துக்காட்டு:
 - abs () – ஒரு எண்ணின் குறியை நீக்கி திருப்பி அனுப்பும்.

```
x=20  
y=-23.2  
print('x = ', abs(x))  
print('y = ', abs(y))  
வெளியீடு  
x = 20  
y = 23.2
```

லாம்ப்டா செயற்கூறுகள்

- பைத்தானில், பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவுக்கு பெயரில்லாத செயற்கூறு அல்லது லாம்ப்டா செயற்கூறு என்று பெயர்.
- பொது வடிவம்
lambda [argument(s)] :expression
- எடுத்துக்காட்டு:

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

தற்சுழற்சி செயற்கூறுகள்

- ஒரு செயற்கூறு தன்னைத் தானே அழைத்தால் அது தற்சுழற்சி என்றழைக்கப்படும்.
- ஒவ்வொரு தற்சுழற்சி செயற்கூற்றிற்கும் அடிப்படை நிபந்தனை கொடுக்கப்பட வேண்டும் இல்லையென்றால் மடக்கு முடிவில்லாமல் இயங்கும்.
- எடுத்துக்காட்டு:

```
def fact(n):  
    if n == 0:  
        return 1  
    else:  
        return n * fact (n-1)  
print (fact (5))
```

2. மாறியின் வரையெல்லைகளை எடுத்துக்காட்டுடன் விளக்குக.

- மாறியின் வரையெல்லை என்பது நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும்.
- வரையெல்லை நடப்பு மாறித் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும்.
- இரண்டு வகையான வரையெல்லைகள் உள்ளன.
 - உள்ளமை வரையெல்லை (Local)
 - முழுதளவிய வரையெல்லை (Global)

உள்ளமை வரையெல்லை

- ஒரு செயற்கூறுவின் உடற்பகுதியின் உள்ளே அறிவிக்கப்படும் மாறி 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 உயிர் பெறும்.
 - முறையான அளபுருக்கள் அதன் செயற்கூறுக்கு உள்ளமையானது ஆகும்.
- எடுத்துக்காட்டு:

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

loc()
வெளியீடு
0
```

முழுதளாவிய வரையெல்லை

- அனைத்து செயற்கூறுகளுக்கும் மேலாக அறிவிக்கப்படும் மாறி முழுதளாவிய மாறி எனப்படும்.
- முழுதளாவிய வரையெல்லை உடைய மாறியை நிரலின் எந்த பகுதியிலும் அணுகமுடியும் முழுதளாவிய வரையெல்லை கொண்ட மாறியை ஒரு செயற்கூறினுள் அணுகுவதற்கு 'global' சிறப்புச் சொல்லானது பயன்படுகிறது.
- பைத்தானின் 'Global' சிறப்புச் சொல்லின் அடிப்படை விதிகள்
 - செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக முழுதளாவிய வரையெல்லை ஆகும். 'global' என்ற சிறப்புச் சொல்லை பயன்படுத்த வேண்டியதில்லை.
 - செயற்கூறினுள் முழுதளாவிய மாறியின் மதிப்பை மாற்றியமைக்க 'global' என்ற சிறப்புச்சொல் பயன்படுத்த வேண்டும்.
 - செயற்கூறுவிற்கு வெளியே 'global' என்ற சிறப்புச்சொல் எந்த விளைவையும் ஏற்படுத்தாது.
- எடுத்துக்காட்டு:

```
x = 0
def add():
    global x
    x = x + 5
    print ("Local X :", x)

add()
print ("Global X :", x)
வெளியீடு
Local X : 5
Global X : 5
```

3. பின்வரும் உள்ளிணைந்த செயற்கூறுகளை விளக்குக.

(அ) id() (ஆ) chr() (இ) round() (ஈ) type() (உ) pow()

செயற்கூறு	விளக்கம்	பொது வடிவம்	எடுத்துக்காட்டு
id()	கொடுக்கப்பட்ட பொருளின் நினைவக முகவரியைத் திருப்பி அனுப்பும்.	id (object)	x=15 print ('address of x is :',id (x)) வெளியீடு address of x is : 135748
chr()	கொடுக்கப்பட்ட ASCII மதிப்பிற்கு யுனிக்கோடு எழுத்தை திருப்பி அனுப்பும்.	chr (i)	c=65 print (chr (c)) வெளியீடு A
round()	கொடுக்கப்பட்ட எண்ணின் அருகாமை முழு எண்ணாக மாற்றி திருப்பி அனுப்பும்.	round (number [ndigits])	x= 17.9 print ('Rounded x value:', round(x)) வெளியீடு:

			Rounded x value: 18
type()	கொடுக்கப்பட்ட பொருளின் தரவின வகையைத் திருப்பி அனுப்பும்.	type (object)	x= 15.2 print (type (x)) வெளியீடு : <class 'float'>
pow()	கொடுக்கப்பட்ட எண்ணின் a ன் அடுக்கு b ன் மதிப்பை அதாவது (a**b) திருப்பி அனுப்பும்.	pow (a,b)	a=5 b=2 c=3.0 print(pow (a,b)) வெளியீடு: 25

4. இரண்டு எண்களின் LCM கண்டுபிடிப்பதற்கான பைத்தான் நிரலை எழுதுக.

```
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
x=int (input("Enter the first number:"))
y=int (input ("Enter the second number:"))
print(lcm(x,y))
```

5. தற்சுழற்சி செயற்கூறு பற்றி எடுத்துக்காட்டு விலக்குக.

- ஒரு செயற்கூறு தன்னைத் தானே அழைத்தால் அது தற்சுழற்சி என்றழைக்கப்படும்.
- எந்தவொரு மடக்கினையும் தற்சுழற்சியாக மாற்ற முடியும்.
- ஒரு நிபந்தனையில் நிறுத்தப்படாவிட்டால், ஒரு செயல்முறை காலவரையின்றி செயல்படும். இந்த செயற்பாடு முடிவிலா சுழற்சி எனப்படும்.
- தற்சுழற்சி செயற்கூறில் கொடுக்கப்படும் நிபந்தனை அடிப்படை நிபந்தனை எனப்படும்.
- ஒவ்வொரு தற்சுழற்சி செயற்கூற்றிற்கும் அடிப்படை நிபந்தனை கொடுக்கப்பட வேண்டும் இல்லையென்றால் மடக்கு முடிவில்லாமல் இயங்கும்.
- தற்சுழற்சி செயற்கூறு எவ்வாறு செயல்படும்
 - தற்சுழற்சி செயற்கூறு வெளிப்புற குறிமுறையிலிருந்து அழைக்கப்படும்.
 - அடிப்படை நிபந்தனை நிறைவேற்றப்பட்டால் நிரலானது சரியான வெளியீடு கொடுத்து வெளியேறும்.
 - இல்லையெனில், செயற்கூறு தேவையான செயற்பாட்டை இயக்கும் மேலும் தற்சுழற்சி முறையில் தன்னைத் தானே அழைத்துக்கொள்ளும்.
- எடுத்துக்காட்டு.

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact (n-1)
print (fact (5))
வெளியீடு
120
```

பாடம் - 8 சரங்கள் மற்றும் சரங்களைக் கையாளுதல்
பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண்)

1. சரம் என்றால் என்ன?

- குறியீடுகளின் அணியை கையாளுவதற்கான ஒரு தரவு இனம் சரம் ஆகும்.
- சரங்கள் என்பன, ஒற்றை, இரட்டை, அல்லது மூன்று மேற்கோள் குறிகளுக்குள் கொடுக்கப்படும்.
- எழுத்து, எண்கள் அல்லது சிறப்பு குறியீடுகளின் சேர்க்கையை கொண்டிருக்கும்.

2. பைத்தானில் சரங்களை மாற்றும் செய்ய முடியுமா?

- பைத்தானில் சர தரவு வகை மாற்ற இயலாதது.
- அதாவது சரத்தரவு வகையை வரையறுத்த பின்பு இயங்கு நேரத்தில் அதை மாற்றியமைக்க முடியாது.

3. பைத்தானில் சரத்தை எவ்வாறு நீக்குவாய் ?

- பைத்தான் சரத்தில் உள்ள ஒரு குறிப்பிட்ட குறியீடுகளை நீக்க அனுமதிக்காது.
- ஆனால் del கட்டளை பயன்படுத்தி ஒரு முழு சர மாறியையும் நீக்க முடியும்.
- எடுத்துக்காட்டு:

del str

4. பின்வரும் பைத்தான் குறிமுறையின் வெளியீடு யாது?

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

வெளியீடு : School School School

5. சரத்தை துண்டாக்குதல் அல்லது பிரித்தல் என்றால் என்ன?

- மூலச்சரத்தில் உள்ள ஒரு துணைச்சரம் (சரத்தின் ஒரு பகுதி) துண்டு எனப்படும்.
- துண்டு அல்லது பிரித்தல் [] செயற்குறியை கொண்டு ஒன்று அல்லது அதற்கு மேற்பட்ட துணைச்சரங்களை / (துண்டுகளாக) மூலச்சரத்திலிருந்து பிரிக்க முடியும்.

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண்)

1. கொடுக்கப்பட்ட வடிவத்தை அச்சிடும் பைத்தான் நிரலை எழுதுக?

```
COMPUTER
COMPUTE
COMPUT
COMPU
COMP
COM
CO
C
```

```
str1="COMPUTER"
index=0
for i in str1:
    print (str1[:index-1])
    index=1
```

2. பின்வருபவற்றை பற்றி தகுந்த எடுத்துக்காட்டுடன் குறிப்பு வரைக

(அ) capitalize() (ஆ) swapcase()

capitalize()

- சரத்தின் முதல் குறியீடுகளை பெரிய எழுத்தாக மாற்ற பயன்படுகிறது.
- ```
>>> city="chennai"
>>> print(city.capitalize())
Chennai
```

swapcase( )

- சரத்தில் உள்ள ஒரு எழுத்து பெரிய எழுத்தாக இருந்தால் அது சிறிய எழுத்தாகவும் நேர் மாறாகவும் திருப்பும்.

```
>>> str1="tAmiL NaDu"
>>> print(str1.swapcase())
TaMlI nAdU
```

3. கீழே கொடுக்கப்பட்டுள்ள பைத்தான் நிரலின் வெளியீடு யாது?

```
str1 = "welcome"
str2 = "to school"
```

வெளியீடு: Weol

```
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

#### 4. format( ) செயற்கூறின் பயன் யாது? எடுத்துக்காட்டு தருக?

- format() செயற்கூறு சரங்களை வடிவமைக்கப் பயன்படும் முக்கிய செயற்கூறாகும்.
- நெளிவு அடைப்புக்குறி { } இட நிரப்பியாக அல்லது புலத்தின் பிரதியீடாக பயன்படுகிறது.
- இது format() செயற்கூறில் பயன்படுத்தப்படும்.
- எடுத்துக்காட்டு:

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
வெளியீடு: Number 1: 34
```

#### 5. பைத்தானில் count( ) செயற்கூறு பற்றி குறிப்பு வரைக.

- ஒரு சரத் தில் கொடுக்கப்பட்ட பரப்பிற்குள் உள்ள துணை சரங்களின் (substring) எண்ணிக்கையை திருப்பும்.
- ஒரு துணை சரம் என்பது ஒற்றை குறியுருவாக கூட இருக்கலாம்.
- தொடரியல்: count(str, beg, end)
- எடுத்துக்காட்டு: >>> str1="Raja Raja Chozhan" >>> print(str1.count('Raja'))  
2

### பகுதி II

#### அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண்)

#### 1. பைத்தானில் பயன்படும் சர செயற்குறிகளை தகுந்த எடுத்துக்காட்டுடன் விளக்குக

- பைத்தான் பின்வரும் செயற்குறிகளை வழங்குகிறது.
- இச்செயற்குறிகள் சரங்களை கையாள உதவுகிறது.

#### இணைப்பு (+)

- இரண்டு அல்லது அதற்கு மேற்பட்ட சரங்களை இணைக்கும் செயல்பாடு சேர்த்தல் அல்லது இணைத்தல் எனப்படும். பைத்தானில் கூட்டல் செயற்குறி சரங்களை இணைக்கப் பயன்படுகிறது.
- எடுத்துக்காட்டு  
>>> "welcome" + "Python"  
'welcomePython'

#### சேர்த்தல் (+=)

- ஏற்கனவே உள்ள சரத்தின் இறுதியில் மேலும் புதிய சரம் அல்லது சரங்களை சேர்க்கும் செயல் சேர்த்தல் எனப்படும்.
- += செயற்குறி ஏற்கனவே உள்ள சரத்தின் இறுதியில் புதிய சரத்தினை சேர்க்க பயன்படுகிறது.
- எடுத்துக்காட்டு

```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
Welcome to Learn Python
```

#### பலமுறை (\*)

- பெருக்கல் செயற்குறி கொடுக்கப்பட்ட சரத்தினை பல தடவைகள் வெளிப்படுத்த பயன்படுகிறது.
- எடுத்துக்காட்டு:

```
>>> str1="Welcome "
>>> print (str1*4)
Welcome Welcome Welcome Welcome
```

**பாடம் - 9 (List, Tuples, Set மற்றும் Dictionary) தொகுப்பு தரவினங்கள்  
பகுதி-ஆ**

**அனைத்து வினாக்களுக்கும் விடையளி**

**(2 மதிப்பெண் )**

**1. பைத்தானில் List என்றால் என்ன?**

- List சரத்தைப் போன்றே "வரிசை முறை தரவினம்" ஆகும்.
- இது சதுர அடைப்புக் குறிக்குள் [ ] அடைக்கப்பட்ட மதிப்புகளின் வரிசைப்படுத்தப்பட்ட தொகுப்பாகும்.
- List-ல் உள்ள ஒவ்வொரு மதிப்பும் உறுப்பு (element) என்றழைக்கப்படுகிறது.

**2. List உறுப்புகளை பின்னோக்கு வரிசையில் தலைகீழாக எவ்வாறு அணுகுவாய்.**

- பைத்தான், List உறுப்புகளுக்கு பின்னோக்கு அல்லது எதிர்மறை, சுட்டெண்களை வழங்குகிறது.
- இதனால் பைத்தான், சுட்டெண்களை எதிர் வரிசையில் பட்டியலிடுகிறது. பைத்தான், List-ன் கடைசி உறுப்பிற்கு-1 முந்தைய உறுப்பிற்கு-2 என்ற சுட்டெண் மதிப்புகளையும் இருத்துகிறது.
- இதுவே பின்னோக்கு சுட்டு என அழைக்கப்படுகிறது.

**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 நீக்குவதற்கு பயன்படுகிறது. | remove( ) செயற்கூறு சுட்டெண் தெரியாத உறுப்புகளை List-லிருந்து நீக்குவதற்கு பயன்படுகிறது. |
| del கூற்று முழு List-ஐ நீக்குவதற்கும் பயன்படுகிறது       | remove ( ) செயற்கூறு ஒரு குறிப்பிட்ட உறுப்பை நீக்க பயன்படுகிறது.                         |

**5. ஒரு Tuples n எண்ணிக்கை உறுப்புகளுடன் உருவாக்குவதற்கான தொடரியலை எழுதுக.**

- tuple\_Name=(E1, E2, E3.....,En)

**6. பைத்தானில் set என்றால் என்ன?**

- Set என்பது ஒரு தொகுப்பு தரவினம் ஆகும்.
- Set என்பது மாறக்கூடிய மற்றும் நகல்கள் இல்லாத வரிசைப்படுத்தப்படாத உறுப்புகளின் தொகுப்பாகும்.
- அதாவது, Set-ல் உள்ள உறுப்புகள் மீண்டும் இடம்பெற முடியாது.

**பகுதி-இ**

**அனைத்து வினாக்களுக்கும் விடையளி**

**(3 மதிப்பெண் )**

**1. List மற்றும் Tuples-ஓப்பிடுக.**

- List-ன் உறுப்புகளை மாற்றலாம் ஆனால் Tuples-ன் உறுப்புகளை மாற்ற முடியாது.
- List-ன் உறுப்புகள் சதுர அடைப்புக் குறிக்குள் அடைக்கப்பட்டிருக்கும். ஆனால், Tuple -ன் உறுப்புகள் வளைவு குறிக்குள் அடைக்கப்பட்டிருக்கும்
- Tuples-ன் மடக்குச் செயல் List-ஐ காட்டிலும் விரைவானது.

**2. Sort( ) பற்றி சிறுகுறிப்பு எழுதுக.**

- List-ல் உள்ள உறுப்புகளை வரிசையாக்கம் செய்கிறது.
- பொது வடிவம்:  
List.sort(reverse=True|False, key=myFunc)
  - reverse, key இரண்டு செயலுருபுகளும் கட்டாயமில்லாதவை
  - reverse ஐ True என பொருத்தினால் இறங்கு வரிசையில் List வரிசையாக்கமாகும்.
  - ஏறுவரிசை தானமைவு வரிசையாகமாகும்.
  - Key=myFunc; "myFunc" - வரிசையாக்க வரண்முறையைக் குறிப்பிடும் பயனர் வரையறுத்த செயற்கூறின் பெயர்.
- எடுத்துக்காட்டு:  
MyList=['Mango', 'Apple', 'Orange']  
MyList.sort( )

```
print(MyList)
MyList.sort(reverse=True)
print(MyList)
```

வெளியீடு

```
['Apple', 'Orange', 'Mango']
['Mango', 'Orange', 'Apple']
```

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 இடையேயான வேறுபாடுகளை எடுத்துக்காட்டுடன் விளக்குக.

| del                                                                                                                                                                       | clear( )                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| del கூற்று முழு List-ஐ நீக்குவதற்கு பயன்படுகிறது.                                                                                                                         | clear( ) செயற்கூறு உறுப்புகளை மட்டும் நீக்கி List-ஐ தொடர்ந்து வைத்திருக்கப் பயன்படுகிறது. |
| Dict = {'Mark1' : 98, 'Mar12' : 86}<br>del Dict['Mark1']<br>print(Dict)<br>del Dict<br>print(Dict)<br>வெளியீடு:<br>{'Mark2': 86}<br>NameError: name 'Dict' is not defined | Dict = {'Mark1' : 98, 'Mar12' : 86}<br>Dict.clear()<br>print(Dict)<br>வெளியீடு:<br>{ }    |

5. பைத்தானின் set செயல்பாடுகளை பட்டியலிடுக.

- ஒட்டு - இரண்டு Set-களை இணைக்க
- வெட்டு - இரண்டு Set-களின் பொதுவான உறுப்புகள்
- வேறுபாடு - இரண்டு Set-களின் வேறுபாடு கண்டறியும்
- சமச்சீரான வேறுபாடு - இரண்டு Set-களின் பொதுவான உறுப்புகள் தவிர்த்து மற்ற உறுப்புகள்

6. List மற்றும் Dictionary இடையேயான வேறுபாடுகள் யாவை ?

| List                                                                 | Dictionary                                                                                            |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| List என்பது வரிசைப்படுத்திய உறுப்புகளின் தொகுப்பாகும்.               | Dictionary ஒரு உறுப்பை (திறவுகோல்) மற்றொரு உறுப்புடன் (மதிப்பு) பொருத்தப் பயன்படும் தரவு அமைப்பாகும். |
| List-ன் சுட்டெண்கள் குறிப்பிட்ட உறுப்பை அணுகுவதற்குப் பயன்படுகின்றன. | Dictionary-ல் திறவுகோல் சுட்டெண்ணைக் குறிக்கிறது.                                                     |
| சுட்டெண் மதிப்புகள் List-ன் மதிப்பை பார்க்க பயன்படுகிறது.            | Dictionary ஒரு மதிப்பை எடுத்துக் கொண்டு மற்றொரு மதிப்பை பார்க்க பயன்படுகிறது.                         |

பகுதி-II

அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண் )

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

append ( )

- append ( ) செயற்கூறு ஒரு உறுப்பை ஏற்கனவே உள்ள List-ல் சேர்க்க பயன்படுகின்றது.

தொடரியல்: List.append (element to be added)

எடுத்துக்காட்டு: >>> MyList=[34, 45, 48]

>>> MyList.append(90)

>>> print(MyList)

[34, 45, 48, 90]

## extend( )

- extend( ) செயற்கூறு ஒன்றுக்கும் மேற்பட்ட உறுப்புகளை ஏற்கனவே உள்ள List-ல் சேர்க்க பயன்படுகின்றது.

தொடரியல்: List.extend ( [elements to be added])  
எடுத்துக்காட்டு: >>> MyList=[34, 45, 48]  
>>> MyList.extend ([71, 32, 29])  
>>> print(MyList)  
[34, 45, 48, 71, 32, 29]

## insert ( )

- insert ( ) செயற்கூறு, List-ன் எந்தவொரு இடத்திலும் ஒரு உறுப்பை சேர்க்கப் பயன்படுகிறது.

தொடரியல்: List.insert(position index, element)  
எடுத்துக்காட்டு: >>> MyList=[34,98,47,90,45]  
>>> print(MyList)  
[34,98,47,90,45]  
>>> MyList.insert(3, 10)  
>>> print(MyList)  
[34,98,47,10,90,45]

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

- range( ) என்பது பைத்தானில் தொடர் மதிப்புகளை உருவாக்கப் பயன்படும் செயற்கூறாகும்.

தொடரியல்: range (start value, end value, step value)

- start value – தொடக்க மதிப்பு. சுழியம் தானமைவு தொடக்க மதிப்பாகும்.
- end value – இறுதி மதிப்பு.
- step value – வெவ்வேறு இடைவெளிகளில் மதிப்புகளை உருவாக்கப் பயன்படுகிறது.

எடுத்துக்காட்டு

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

வெளியீடு 2 4 6 8 10

## தொடர் மதிப்புகளுடன் List-ஐ உருவாக்குதல்

- range( ) செயற்கூறை பயன்படுத்தி நீங்கள் தொடர் மதிப்புகளுடன் கூடிய List-ஐ உருவாக்கலாம்.
- range( ) செயற்கூறின் விடையை List ஆக மாற்றுவதற்கு, List( ) செயற்கூறு பயன்படுகிறது.

தொடரியல்: List\_Varibale = List ( range ( ) )

எடுத்துக்காட்டு >>> Even\_List = List(range(2,11,2))  
>>> print(Even\_List)  
[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 = (("Ram", 98.7), ("Ravi", 97.5), ("Raju ", 95.3), ("Arun", 93.8))
for i in Toppers:
 print(i)
```

வெளியீடு

```
('Ram', 98.7)
('Ravi', 97.5)
('Raju', 95.3)
('Arun', 93.8)
```

## 4. பைத்தானிலுள்ள பல்வேறு set செயல்பாடுகளை பொருத்தமான எடுத்துக்காட்டுகளுடன் விளக்குக.

## ஒட்டு

- இது இரண்டு அல்லது அதற்கு மேற்பட்ட Set-களின் அனைத்து உறுப்புகளையும் உள்ளடக்கும்.
- பைத்தானில் | என்ற செயற்குறியும் மற்றும் Union() செயற்கூறும் இரண்டு Set களை இணைக்கப் பயன்படுகிறது.

```
எடுத்துக்காட்டு: set_A={2,4,6,8}
 set_B={'A', 'B', 'C', 'D'}
 set_U=set_A|set_B (அல்லது) set_U=set_A.union(set_B)
 print(set_U)
 வெளியீடு: {2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

## வெட்டு

- இது இரண்டு Setகளின் பொதுவான உறுப்புகளை உள்ளடக்கியது
- பைத்தானில் & செயற்குறியும் மற்றும் intersection() செயற்கூறும் இரண்டு Set களை வெட்டுவதற்கு பயன்படுகிறது.

```
எடுத்துக்காட்டு: set_A={'A', 2, 4, 'D'}
 set_B={'A', 'B', 'C', 'D'}
 print(set_A & set_B) (அல்லது) print(set_A.intersection(set_B))
 வெளியீடு: {'A', 'D'}
```

## வேறுபாடு

- இது முதல் Set(A) ல் உள்ள அனைத்து உறுப்புகளையும் உள்ளடக்கிய இரண்டாவது Set-ஐ தவிர்க்கிறது.
- பைத்தானில் -(minus) செயற்குறியும் மற்றும் difference() செயற்கூறும் வேறுபாட்டு செயற்பாட்டிற்காக பயன்படுகிறது.

```
எடுத்துக்காட்டு: set_A={'A', 2, 4, 'D'}
 set_B={'A', 'B', 'C', 'D'}
 print(set_A - set_B) (அல்லது) print(set_A.difference(set_B))
 வெளியீடு: {2, 4}
```

## சமச்சீரான வேறுபாடு

- இது இரண்டு Set-ல் உள்ள பொதுவான உறுப்புகளை மட்டும் தவிர்த்து மற்ற அனைத்து உறுப்புகளையும் உள்ளடக்கியது.
- பைத்தானில் ^ செயற்குறியும் மற்றும் Symmetric\_difference() செயற்கூறும் சமச்சீரான வேறுபாட்டை கண்டறிய பயன்படுகிறது.

```
எடுத்துக்காட்டு: set_A={'A', 2, 4, 'D'}
 set_B={'A', 'B', 'C', 'D'}
 print(set_A ^ set_B) (அல்லது) print(set_A.symmetric_difference(set_B))
 வெளியீடு: {2, 4, 'B', 'C'}
```

பாடம் - 10 பைத்தான் இனக்குழுக்கள் மற்றும் பொருள்கள் பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண் )

1. இனக்குழு என்றால் என்ன ?

- இனக்குழு என்பது பொருளின் வார்ப்புரு ஆகும்.
- இனக்குழு மாறிகள் மற்றும் வழிமுறைகள் சேர்ந்து இனக்குழுவின் உறுப்புகள் எனப்படும்.

2. சான்றுருவாக்கல் என்றால் என்ன ?

- இனக்குழு உருவாக்கப்பட்ட பின் அந்த இனக்குழுவின் பொருள் அல்லது சான்றுரு உருவாக்கப்பட வேண்டும்.
- பொருளை உருவாக்கும் இந்த முறைக்கு சான்றுருவாக்கல் என்று பெயர்.

3. பின்வரும் நிரலின் வெளியீடு என்ன ?

```
class Sample:
 __num=10
 def disp(self):
```

```

print(self.__num)
S=Sample()
S.disp()
print(S.__num)

```

வெளியீடு:

```

10
Attribute Error: 'Sample' object has no
attribute '__num'

```

#### 4. பைத்தானில் ஆக்கியை எவ்வாறு உருவாக்குவாய்?

- பைத்தானில், "init" என்னும் சிறப்பு செயற்கூறு ஆக்கியாக செயல்படுகிறது.
- இது இரட்டை அடிக்கீறில் (UnderScore) தொடங்கி இரட்டை அடிக்கீறலுடன் முடிய வேண்டும்.
- பொருள் உருவாக்கப்பட்டவுடன் தானாகவே இயக்கப்படும்.
- இந்த ஆக்கி செயற்கூறை அளபுருக்களுடனோ அல்லது இல்லாமலோ வரையறுக்கலாம்.

பொது வடிவம்: def \_\_init\_\_(self, [args.....]):  
<statements>

எடுத்துக்காட்டு class Sample:

```

def __init__(self, num):
 print("Constructor of class Sample...")
 self.num=num
 print("The value is :", num)

```

S=Sample(10)

#### 5. அழிப்பியின் நோக்கம் என்ன ?

- இனக்குழுவில் உருவாக்கப்பட்ட பொருளின் பயன்பாடு முடிவுக்கு வரும் போது அழிப்பி என்னும் சிறப்பு செயற்கூறு தானாகவே இயக்கப்படும்.
- \_\_del\_\_( ) செயற்கூறு அழிப்பியாகப் பயன்படுத்தப்படுகிறது.

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண் )

#### 1. இனக்குழு உறுப்புகள் என்றால் என்ன ? அதனை எவ்வாறு வரையறுப்பாய் ?

- இனக்குழுக்குள்ளே வரையறுக்கப்படும் மாறிகள் இனக்குழு மாறிகள் என்றும் செயற்கூறுகள் வழிமுறைகள் என்றும் அழைக்கப்படும்.
- இனக்குழு மாறிகள் மற்றும் வழிமுறைகள் சேர்ந்து இனக்குழுவின் உறுப்புகள் எனப்படும்.
- எடுத்துக்காட்டு:

```

class Sample:
 x= 10 # இனக்குழு மாறி
 def disp(self) : # வழிமுறை
 print(Sample.x)
s=Sample()
s.disp()

```

#### 2. இரண்டு private இனக்குழு மாறிகளுடன், வழிமுறையை பயன்படுத்தி கூட்டுத்தொகை sum அச்சிடும் இனக்குழுவை வரையறுக்கவும்.

```

class sample:
 def __init__(self,n1,n2):
 self.__n1=n1
 self.__n2=n2
 def sum(self):
 print(self.__n1 + self.__n2)
s=sample(12,14)
s.sum()
வெளியீடு: 26

```

#### 3. கொடுக்கப்பட்ட வெளியீட்டை பெற பின்வரும் 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()
```

வெளியீடு : Fruit 1 = Apple, Fruit 2 = Mango  
விடை : மேற்கண்ட வெளியீட்டைப் பெற del F.display என்ற வரியை நிரலில் இருந்து நீக்கவும்

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 அழிப்பிகளை எவ்வாறு வரையறுப்பாய் ?

- ஆக்கி வரையறுப்பின் பொது வடிவம்
 

```
def __init__(self, [args]) :
 <statements>
```
- அழிப்பி வரையறுப்பின் பொது வடிவம்
 

```
def __del__(self) :
 <statements>
```

பகுதி-ஈ

அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண் )

1. ஆக்கி மற்றும் அழிப்பி பற்றி பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக .

ஆக்கிகள்

- பைத்தானில், "init" என்னும் சிறப்பு செயற்கூறு ஆக்கியாக செயல்படுகிறது.
- இது இரட்டை அடிக்கீறில் (UnderScore) தொடங்கி இரட்டை அடிக்கீறலுடன் முடிய வேண்டும்.
- இந்த செயற்கூறு சாதாரண செயற்கூறுவைப் போல செயல்படும். ஆனால் பொருள் உருவாக்கப்பட்டவுடன் தானாகவே இயக்கப்படும். இந்த ஆக்கி செயற்கூறை அளபுருக்களுடனோ அல்லது இல்லாமலோ வரையறுக்கலாம்.
- பொது வடிவம்:
 

```
def __init__(self, [args]):
 <statements>
```
- எடுத்துக்காட்டு:
 

```
class Sample:
 def __init__(self, num):
 print("Constructor of class Sample...")
 self.num=num
 print("The value is :", num)
S=Sample(10)
```

அழிப்பிகள்

- இனக்குழுவில் உருவாக்கப்பட்ட பொருளின் பயன்பாடு முடிவுக்கு வரும் போது அழிப்பி என்னும் சிறப்பு செயற்கூறு தானாகவே இயக்கப்படும்.
- \_\_del\_\_( ) செயற்கூறு அழிப்பியாகப் பயன்படுத்தப்படுகிறது.
- பொது வடிவம்:
 

```
def __del__(self) :
 <statements>
```
- எடுத்துக்காட்டு:
 

```
class sample:
 def __init__(self,n1):
 self.__n1=n1
 print("Constructor of class Sample...")
 def __del__(self):
 print("Destructor of class Sample...")
s=sample(5)
```

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண் )

1. தரவுத்தள மேலாண்மை அமைப்பிற்கு சில எடுத்துக்காட்டுகளைக் குறிப்பிடுக.

- Foxpro
- dbase

2. RDBMS-ன் சில எடுத்துக்காட்டுகளைப் பட்டியலிடுக.

- SQL server
- Oracle
- MySQL
- MariaDB
- SQLite
- MS-Access

3. தரவு நிலைத்தன்மை என்றால் என்ன?

- தரவு நிலைத்தன்மை என்பது தரவுத்தளத்தில் அனைத்து இடங்களிலும் ஒத்த மதிப்புடைய தரவுகளாகும்.
- தொடர்ச்சியாக புதுப்பிக்கப்படும் மற்றும் பராமரிக்கப்படும் நேரடி தரவின் நிலைத்தன்மையை கையாளுவதே தரவு நிலைத்தன்மை எனப்படும்.

4. படிநிலை மற்றும் வலையமைப்பு தரவு மாதிரிக்கும் இடையே உள்ள வேறுபாடு என்ன?

- படிநிலை மாதிரியில், ஒரு குழந்தை பதிவு ஒரே ஒரு பெற்றோர் முனையத்தை மட்டுமே கொண்டிருக்கும்.
- வலையமைப்பு மாதிரியில் ஒரு குழந்தைக்கு பல பெற்றோர் முனையங்கள் இருக்கலாம். இது தரவை பலவற்றிலிருந்து பல உறவுநிலையை குறிப்பிடுகிறது
- வலையமைப்பு மாதிரி எளிமையாகவும், விரைவாகவும் தரவுகளை அணுக பயன்படுகிறது

5. இயல்பாக்கம் என்றால் என்ன?

- இயல்பாக்கம் என்பது தரவு மிகைமைக் குறைக்கிறது மற்றும் தரவு நிலைப்பாட்டை அதிகரிக்கிறது.

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண் )

1. Select மற்றும் Project செயற்பாடுகளின் வேறுபாடுகள் யாவை?

| Select                                                                               | Project                                                                |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| ஒரு நிபந்தனையின் அடிப்படையில் துணைத் தொகுதியை tuples களுடன் தேர்ந்தெடுக்க உதவுகிறது. | குறிப்பிடப்பட்டுள்ள உள்ளீட்டு தொடர்புகளின் பண்புக்கூறுகளை நீக்குகிறது. |
| நிபந்தனைகளை நிறைவேற்ற tuples களை Select வடிவாக்கிகிறது.                              | இந்த செயற்பாடு கிடக்கை துணைத் தொகுதியின் ஒப்பீடுகளை வரையறுக்கிறது.     |
| symbol : $\sigma$                                                                    | symbol : $\Pi$                                                         |

2. DBA வின் பணி என்ன?

- DBA என்பவர் முழு தரவுத்தள மேலாண்மை அமைப்பையும் நிர்வகிப்பவர் ஆவார்.
- இவர் DBMS பாதுகாப்பு, உரிமங்களை நிர்வகித்தல், பயனர் கணக்குகள் மற்றும் அணுகல்களை நிர்வகித்தல் போன்றவற்றை கவனித்துக் கொள்கிறார்.

3. கார்டசியன் பெருக்கலை பொறுத்தமான எடுத்துக்காட்டுடன் விளக்குக.

- இரண்டு தொடர்புகளை சேர்க்க குறுக்குப் பெருக்கல் வழிவகுக்கிறது. இதன் விடை இரண்டு தொடர்புகளின் இணைப்பைக் கொண்டுள்ளது.
- இரண்டு தொடர்புகளிலிருந்து நெடுக்கைகளை ஒன்று சேர்க்க பயன்படுகிறது.

| Student |      |
|---------|------|
| Rollno  | Name |
| 101     | Ram  |
| 102     | Raj  |

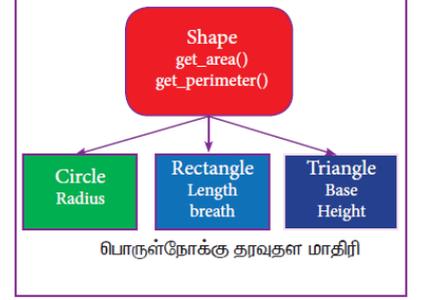
| Course  |      |       |
|---------|------|-------|
| Crollno | Cno  | Cname |
| 101     | C120 | JAVA  |
| 102     | C121 | C++   |

Student X Course

| Rollno | Name | Crollno | Cno  | CName |
|--------|------|---------|------|-------|
| 101    | Ram  | 101     | C120 | JAVA  |
| 101    | Ram  | 102     | C121 | C++   |
| 102    | Raj  | 101     | C120 | JAVA  |
| 102    | Raj  | 102     | C121 | C++   |

4. பொருள் மாதிரியை எடுத்துக்காட்டுடன் விளக்குக.

- இந்த மாதிரி தரவை பொருள்கள், பண்புக்கூறுகள், வழிமுறைகள், இனக்குழு மற்றும் மரபுரிமம் போன்ற வழிகளில் சேமிக்கிறது.
- இது மிகவும் சிக்கலான பயன்பாடுகளை கையாள்கிறது.
- இது கோப்பு மேலாண்மை அமைப்பில் பயன்படுகிறது.
- நிகழ் உலக பொருள்கள், பண்புக்கூறுகள், பண்பியல்புகளை குறிப்பிடுகிறது மற்றும் தெளிவான கூறுநிலை அமைப்பை வழங்குகிறது. இதில் ஏற்கனவே உள்ள குறிமுறையை எளிதாகப் பராமரிக்கவும், மாற்றி அமைக்கவும் முடியும்.



5. DBMS-ன் பல்வேறு வகையான பயனர்களைப் பற்றி குறிப்பு வரைக.

தரவுத்தள நிர்வாகிகள்

- முழு தரவுத்தள மேலாண்மை அமைப்பையும் நிர்வகிப்பவர்.

பயன்பாட்டு நிரலர்கள் அல்லது மென்பொருள் உருவாக்குபவர்கள்

- DBMSன் பகுதிகளை உருவாக்குதல் மற்றும் வடிவமைத்தல்

இறுதி நிலைப்பயனர்

- தரவை சேமித்தல், திரும்ப பெறுதல், புதுப்பித்தல் மற்றும் நீக்குதல் போன்ற பல்வேறு செயல்களை கையாள்கிறார்கள் .

தரவுத்தள வடிவமைப்பாளர்கள்:

- சரியான கட்டமைப்பை தேர்வு செய்து தரவை குறிப்பிடுவதற்கும் சேமிப்பதற்குமான பொறுப்பாளர்கள் ஆவர்

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி

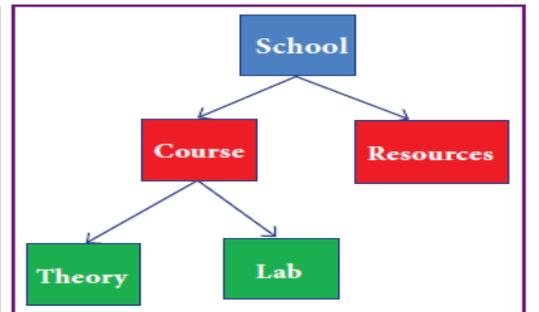
(5 மதிப்பெண் )

1. தரவு மாதிரியின் பல்வேறு வகைகளை விளக்குக.

- தரவு மாதிரியின் பல்வேறு வகைகள் பின்வருமாறு
  1. படிநிலை தரவுதள மாதிரி
  2. உறவுநிலை தரவுதள மாதிரி
  3. வலையமைப்பு தரவுதள மாதிரி
  4. ER தரவுதள மாதிரி
  5. பொருள்நோக்கு தரவுதள மாதிரி

1. படிநிலை தரவுதள மாதிரி

- தரவு எளிமையான மரக்கிளை அமைப்பில் குறிப்பிடப்படுகிறது.
- இது ஒன்றிலிருந்து பல உறவு நிலையை குறிக்கிறது.
- அதாவது பெற்றோர் - குழந்தை உறவுநிலை.
- ஒரு குழந்தைக்கு ஒரு பெற்றோர் இருப்பர். ஆனால், ஒரு பெற்றோருக்கு பல குழந்தைகள் இருக்கலாம்.



## 2. உறவுநிலை தரவுதள மாதிரி

- தரவுகளின் அடிப்படை கட்டமைப்பு அட்டவணைகள் (உறவுகள்) ஆகும்.
- ஒரு குறிப்பிட்ட வகையை சார்ந்த அனைத்து தகவல்களும் அவ்வட்டவணையின் வரிசைகளில் சேமிக்கப்படுகின்றன.
- ஒரு உறவுநிலை திறவுகோல் குறிப்பிட்ட வரிசையிலான தரவுகளை தனித்தன்மையுடன் குறிக்கும் ஒரு பண்புக்கூறு ஆகும்.

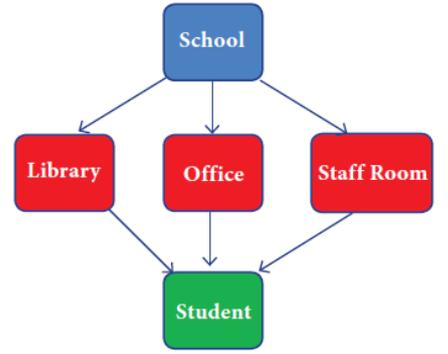
| Stu_id | Name   | Age | Subj_id | Name   | Teacher      |
|--------|--------|-----|---------|--------|--------------|
| 1      | Malar  | 17  | 1       | C++    | Kannan       |
| 2      | Suncar | 16  | 2       | Php    | Ramakrishnan |
| 3      | Velu   | 16  | 3       | Python | Vidhya       |

| Stu_id | Subj_id | Marks |
|--------|---------|-------|
| 1      | 1       | 92    |
| 1      | 2       | 89    |
| 3      | 2       | 96    |

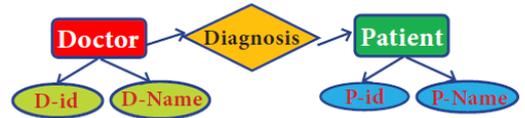
## 3. வலையமைப்பு தரவுதள மாதிரி

- படிநிலை தரவுத்தள மாதிரியின் விரிவாக்கப்பட்ட அமைப்பாகும்.
- ஒரு குழந்தைக்கு பல பெற்றோர் முனையங்கள் இருக்கலாம்.
- இது தரவை பலவற்றிலிருந்து பலவற்றிற்கு உறவு நிலையை குறிப்பிடுகிறது
- எளிமையாகவும், விரைவாகவும் தரவுகளை அணுக பயன்படுகிறது.



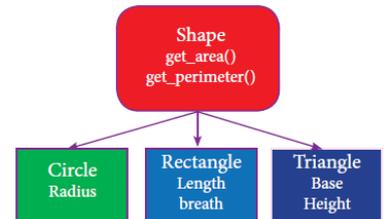
## 4. E R தரவுதள மாதிரி

- பொருளை உருப்படியாகவும் அதன் பண்புகளை, பண்புக் கூறுகளாகவும் பிரித்து உறவுநிலை உருவாக்கப்படுகிறது.
- தரவின் விளக்கப்படத்தை வடிவமைக்க மிகவும் எளிமையாக உள்ளது.
- ER மாதிரியைக் கொண்டு நிரலர் அமைப்பை எளிமையாக புரிந்து கொள்ள முடியும்.
- செவ்வகம் உருப்படிகளைக் குறிக்கிறது.
- நீள்வட்டம் பண்புக்கூறுகளைக் குறிக்கிறது.
- டைமண்ட் ER படங்களின் உறவுநிலையைக் குறிப்பிடுகிறது.



## 5. பொருள்நோக்கு தரவுதள மாதிரி

- இந்த மாதிரி தரவை பொருள்கள், பண்புக்கூறுகள், வழிமுறைகள், இனக்குழு மற்றும் மரபரிமம் போன்ற வழிகளில் சேமிக்கிறது.
- இது மிகவும் சிக்கலான பயன்பாடுகளை கையாள்கிறது.
- இது கோப்பு மேலாண்மை அமைப்பில் பயன்படுகிறது.
- நிகழ் உலக பொருள்கள், பண்புக்கூறுகள், பண்பியல்புகளை குறிப்பிடுகிறது மற்றும் தெளிவான கூறுநிலை அமைப்பை வழங்குகிறது. இதில் ஏற்கனவே உள்ள குறிமுறையை எளிதாகப் பராமரிக்கவும், மாற்றி அமைக்கவும் முடியும்.

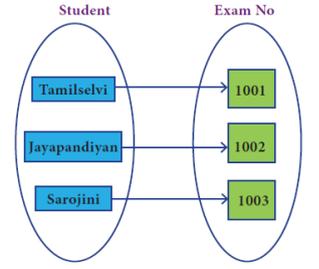


## 2. உறவுநிலையின் வகைகளை விவரி.

- தரவுத்தளத்தில் பயன்படும் உறவுநிலைகளின் வகைகள்.
  1. ஒன்றுடன் ஒன்று உறவுநிலை
  2. ஒன்றுடன் பல உறவுநிலை
  3. பலவற்றுடன் ஒன்று உறவுநிலை
  4. பலவற்றுடன் பல உறவுநிலை

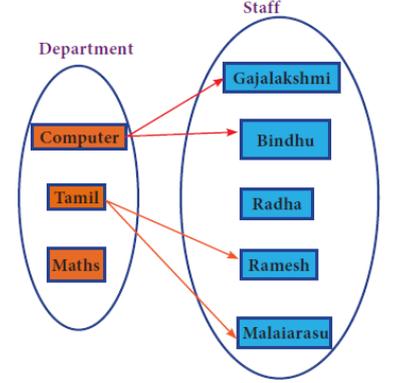
### ஒன்றுடன் ஒன்று உறவுநிலை

- ஒரு அட்டவணை யின் ஒரு பதிவு மற்றொரு அட்டவணை யின் ஒரு பதிவுடன் மட்டுமே இணைக்கப்பட்டிருக்கும்.
- எடுத்துக்காட்டு ஒரு மாணவனுக்கு ஒரு தேர்வெண் மட்டுமே இருக்க முடியும்.



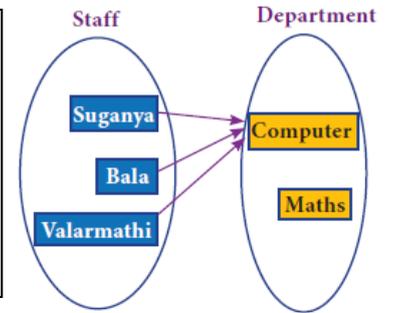
### ஒன்றுடன் பல உறவுநிலை

- அட்டவணை A-யின் ஒரு பதிவு அட்டவணை B யின் பல பதிவுகளுடன் தொடர்பு கொண்டிருக்கலாம். ஆனால் அட்டவணை B யின் ஒரு பதிவு அட்டவணை A-யின் ஒரு பதிவுடன் மட்டுமே தொடர்பு கொண்டிருக்க முடியும்.
- எடுத்துக்காட்டு: ஒரு துறை பல ஊழியர்களைக் கொண்டுள்ளது.



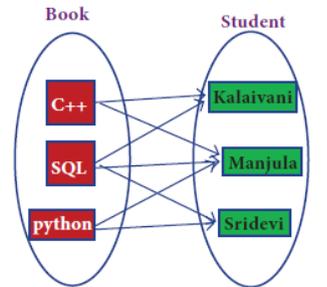
### பலவற்றுடன் ஒன்று உறவுநிலை

- இந்த உறவுநிலையில் பல பொருள்கள் வேறு ஒரேயொரு பொருள்களுடன் மட்டுமே தொடர்பு கொண்டிருக்கும்.
- எடுத்துக்காட்டு: ஒரு துறையில் வேலை செய்யும் ஊழியர்களின் எண்ணிக்கையில், ஊழியர்கள் அட்டவணை யின் பல பதிவுகள் துறை அட்டவணை யின் ஒரு பதிவுடன் மட்டுமே தொடர்பு படுத்தப்பட்டிருக்கும்.



### பலவற்றுடன் பல உறவுநிலை

- இந்த உறவுநிலையில் ஒரு அட்டவணை யில் உள்ள பல பதிவுகள் மற்றொரு அட்டவணை யில் உள்ள பல பதிவுகளுடன் தொடர்பு கொண்டிருக்கும்.
- எடுத்துக்காட்டு: புத்தகங்கள் மற்றும் மாணவன்.
  - நூலகத்தில் உள்ள பல புத்தகங்கள் பல மாணவர்களுக்கு வழங்கப்படும்.



### 3. DBMS மற்றும் RDBMS வேறுபடுத்துதல்.

| ஒப்பீட்டு அடிப்படை | DBMS                                                           | RDBMS                                                               |
|--------------------|----------------------------------------------------------------|---------------------------------------------------------------------|
| விரிவாக்கம்        | Database Management System                                     | Relational DataBase Management System                               |
| தரவு சேமிப்பு      | வழிசெலுத்தல் மாதிரி (ஒன்றோடொன்று இணைக்கப்பட்ட பதிவுகளின் தரவு) | உறவுநிலை மாதிரி (அட்டவணை யில் வரிசை மற்றும் நெடுவரிசை உள்ள தரவுகள்) |
| மிகைமைத் தரவு      | இடம் பெற்றுள்ளது                                               | இடம் பெறவில்லை                                                      |
| இயல்பாக்கம்        | செய்ய இயலாது                                                   | இயல்பாக்கத்தை பயன்படுத்துகிறது.                                     |
| தரவு அணுகல்        | அதிக நேரத்தை எடுத்துக்கொள்கிறது.                               | DBMS உடன் ஒப்பிடும் போது வேகமானது.                                  |

|                                  |                                                |                                                        |
|----------------------------------|------------------------------------------------|--------------------------------------------------------|
| திறவுகோல்கள் மற்றும் குறியீடுகள் | பயன்படுத்தப்படவில்லை                           | உறவுநிலையை உருவாக்குவதற்குப் பயன்படுகிறது.             |
| பரிவர்த்தனை மேலாண்மை             | திறமையற்றது, பிழைகளைக் கொண்டது பாதுகாப்பற்றது. | திறமையானது மற்றும் பாதுகாப்பானது.                      |
| பரவல் தகவல்தளம்                  | ஒத்துழைக்காது                                  | ஒத்துழைக்கும்                                          |
| எடுத்துக்காட்டு                  | Dbase, FoxPro.                                 | SQL server, Oracle, MySQL, MariaDB, SQLite, MS–Access. |

**4. ஒட்டுதல், வெட்டுதல், வேறுபாடு மற்றும் கார்டிசியன் பெருக்கல் போன்றவற்றை பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.**

ஒட்டுதல் (Symbol :  $\cup$ )

- இது A அல்லது B அட்டவணையில் உள்ள அனைத்து tuples-களையும் உள்ளடக்கியது.
- இது நகல்களையும் நீக்குகிறது.
- தொகுதி A ஒட்டுதல் தொகுதி B என்பது  $A \cup B$

| Table A |         | Table B |              | Table $A \cup B$ |              |
|---------|---------|---------|--------------|------------------|--------------|
| Studno  | Name    | Studno  | Name         | Studno           | Name         |
| cs1     | Kannan  | cs1     | Kannan       | cs1              | Kannan       |
| cs3     | Lenin   | cs2     | GowriShankar | cs2              | GowriShankar |
| cs4     | Padmaja | cs3     | Lenin        | cs3              | Lenin        |

வேறுபாடு ( Symbol :  $-$  )

- இது A மற்றும் B அட்டவணைகளை ஒப்பிட்டு A அட்டவணையில் உள்ள வேறுபட்ட tuples களை மட்டும் தருகிறது. B அட்டவணையைத் தவிர்க்கிறது.
- A பண்புக்கூறின் பெயர் B பண்புக்கூறின் பெயரோடு பொருந்த வேண்டும்.

| Table A - B |         |
|-------------|---------|
| cs4         | Padmaja |

வெட்டு (symbol :  $\cap$ )  $A \cap B$

- இது A மற்றும் B அட்டவணைகளில் பொதுவாக உள்ள அனைத்து tuples களையும் உள்ளடக்கியது.

| $A \cap B$ |        |
|------------|--------|
| cs1        | Kannan |
| cs3        | Lenin  |

கார்டிசியன் பெருக்கல் (Symbol :  $\times$  )

- இரண்டு தொடர்புகளை சேர்க்க குறுக்குப் பெருக்கல் வழிவகுக்கிறது.
- இதன் விடை இரண்டு தொடர்புகளின் இணைப்பைக் கொண்டுள்ளது.
- $A \times B$  என்பது A times B, இங்கு A தொடர்புகள் மற்றும் B தொடர்புகள் என்பன வேறுபட்ட பண்புக்கூறுகளாகும்.
- இந்த வகை செயற்பாடுகள் இரண்டு தொடர்புகளிலிருந்து நெடுக்கைகளை ஒன்று சேர்க்க பயன்படுகிறது.

| Student |      |
|---------|------|
| Rollno  | Name |
| 101     | Ram  |
| 102     | Raj  |

| Course  |      |       |
|---------|------|-------|
| Crollno | Cno  | Cname |
| 101     | C120 | JAVA  |
| 102     | C121 | C++   |

Student X Course

| Rollno | Name | Crollno | Cno  | CName |
|--------|------|---------|------|-------|
| 101    | Ram  | 101     | C120 | JAVA  |
| 101    | Ram  | 102     | C121 | C++   |
| 102    | Raj  | 101     | C120 | JAVA  |
| 102    | Raj  | 102     | C121 | C++   |

5. RDBMS ன் பண்பியல்புகளை விளக்குக.

|                                                |                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. தரவுகளைகையாளும் திறன்                       | <ul style="list-style-type: none"> <li>உறவுநிலை தரவுதள மேலாண்மை அமைப்பானது தரவு தளத்தில் தரவுகளை கையாளும் வசதியை வழங்குகிறது.</li> </ul>                                                                                                                                                                                                |
| 2. மிகைமை குறைத்தல்                            | <ul style="list-style-type: none"> <li>RDBMS தரவுகளை பிரித்தாளும் இயல்பு நிலையைப்பின்பற்றி தேவையற்ற தரவுகள் மீண்டும் இடம் பெறுவதைக் குறைக்கிறது.</li> </ul>                                                                                                                                                                             |
| 3. தரவுகளின் நிலைத்தன்மை                       | <ul style="list-style-type: none"> <li>தொடர்ச்சியாக புதுப்பிக்கப்படும் மற்றும் பராமரிக்கப்படும் நேரடி தரவின் நிலைத்தன்மையை பராமரிப்பது ஒரு சவாலாக மாறும்.</li> <li>ஆனால் RDBMS தானாகவே அதை கையாளுகிறது.</li> </ul>                                                                                                                      |
| 4. பல பயனர்கள் மற்றும் ஒத்த அணுகுதலை ஆதரித்தல் | <ul style="list-style-type: none"> <li>பல பயனர்களை ஒரே நேரத்தில் ஒரு தரவின் மீது வேலை செய்ய அனுமதிக்கிறது.</li> <li>இருப்பினும் தரவின் நிலைத்தன்மைப் பராமரிப்பை கையாளுகிறது.</li> </ul>                                                                                                                                                 |
| 5. வினவல் மொழி                                 | <ul style="list-style-type: none"> <li>எளிய வினவல் மொழியை பயனருக்கு வழங்குகிறது.</li> <li>இதைப்பயன்படுத்தி தரவுத்தளத்திலிருந்து தரவுகளை எளிமையாக பெறவும், சேர்க்கவும், நீக்கவும், புதுப்பிக்கவும் முடியும்.</li> </ul>                                                                                                                  |
| 6. பாதுகாப்பு                                  | <ul style="list-style-type: none"> <li>தரவின் பாதுகாப்பை கவனித்துக்கொள்கிறது.</li> <li>அங்கீகாரமற்ற அணுகுதலில் இருந்து தரவுகளுக்கு பாதுகாப்பை வழங்குகிறது.</li> <li>பயனர் கணக்குகளை நாம் உருவாக்க முடியும்.</li> <li>இதனைப் பயன்படுத்தி பயனர் அணுகுதலை கட்டுப்படுத்தி நம்முடைய தரவுகளை எளிமையாக பாதுகாத்துக் கொள்ள முடியும்.</li> </ul> |
| 7. RDBMS பரிவர்த்தனைகளை ஆதரித்தல்              | <ul style="list-style-type: none"> <li>இது நிகழ் உலக பயன்பாடுகளில் தரவு ஒருமைப்பாட்டை கையாள மற்றும் நிர்வகிக்க நம்மை அனுமதிக்கிறது.</li> <li>இங்கு பல்புரியாக்கம் விரிவாக பயன்படுகிறது.</li> </ul>                                                                                                                                      |

பாடம்-12 வினவல் அமைப்பு மொழி

பகுதி - ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண் )

1. 18 வயதிற்கும் குறைவாக உள்ள அனைத்து மாணவர்களின் தரவினை வரிசைப்படி தெரிவு செய்யும் ஒரு வினவலை எழுதுக.

SELECT \* FROM student WHERE age<18 ORDER BY name;

2. Unique மற்றும் Primary Key கட்டுப்பாடுகளை வேறுபடுத்துக

| Unique Key                                                                                        | Primary Key                                                                                                                  |
|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| குறிப்பிட்ட நெடுவரிசைகளில் எந்த ஒரு வரிசைகளும் ஒரே மதிப்பை கொண்டிருக்காது என்பதை உறுதி செய்கிறது. | தரவுத்தளத்திலுள்ள ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட ஒரு புலத்தினை primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்க வேண்டும். |
| ஒன்றுக்கும் மேற்பட்ட புலங்களை Unique key ஆக பயன்படுத்தலாம்                                        | ஒரு புலத்தை மட்டுமே primary key ஆக குறிப்பிட முடியும்.                                                                       |

|                                                                         |                                                                                                 |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| NOT NULL என்று அறிவிக்கப்பட்ட புலங்களுக்கு மட்டுமே பயன்படுத்த முடியும். | வெற்று (NULL) மதிப்புகளை அனுமதிக்காது என்பதால், NOT NULL என்ற கட்டுப்பாட்டையும் கொண்டிருக்கும். |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|

3. அட்டவணை கட்டுப்பாட்டிற்கும், நெடுவரிசை கட்டுப்பாட்டிற்கும் உள்ள வேறுபாடுகளை எழுதுக?

|                                                                                        |                                                                              |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| அட்டவணை கட்டுப்பாடு                                                                    | நெடுவரிசை கட்டுப்பாடு                                                        |
| அட்டவணை கட்டுப்பாடு ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளுக்கு பயன்படுத்த முடியும். | தனிப்பட்ட நெடுவரிசைக்கு மட்டுமே நெடுவரிசை கட்டுப்பாட்டை பயன்படுத்த முடியும். |

4. எந்த SQL கூறு, அட்டவணையை உருவாக்கவும், அவற்றில் மதிப்புகளை சேர்க்கவும் அனுமதிக்கும்?

|                           |              |                                |
|---------------------------|--------------|--------------------------------|
| SQL கூறு                  | கட்டளை       | பயன்பாடு                       |
| தரவு வரையறை மொழி (DDL)    | CREATE TABLE | அட்டவணையை உருவாக்க             |
| தரவு கையாளுதல் மொழி (DML) | INSERT       | அட்டவணையில் மதிப்புகளை சேர்க்க |

5. SQL மற்றும் MySQLக்கு உள்ள வேறுபாடுகள் யாவை ?

- SQL : தரவுத்தளங்களை அணுகுவதற்கான வினவல் அமைப்பு மொழி
- MySQL: MySQL ஒரு உறவுநிலை தரவுத்தள மேலாண்மை அமைப்பு (RDBMS)

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண் )

1. கட்டுப்பாடு என்றால் என்ன ? Primary Key கட்டுப்பாடு பற்றி சிறுகுறிப்பு வரைக.

- கட்டுப்பாடுகள் என்பது ஒரு புலம் அல்லது புலங்களின் தொகுதிக்கு பயன்படுத்தக்கூடிய நிபந்தனையை குறிக்கும்.

Primary Key கட்டுப்பாடு

- தரவுத்தளத்திலுள்ள ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட ஒரு புலத்தினை primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்க வேண்டும்.
- ஒரு புலத்தை மட்டுமே primary key ஆக குறிப்பிட முடியும்.
- வெற்று (NULL) மதிப்புகளை அனுமதிக்காது என்பதால் NOT NULL என்ற கட்டுப்பாட்டையும் கொண்டிருக்க வேண்டும்.

2. ஒரு புதிய புலத்தை சேர்ப்பதன் மூலம் மாணவர் அட்டவணை கட்டமைப்பை மாற்றி அமைக்க ஒரு SQL கூற்றை எழுதுக.

ALTER TABLE student ADD address char(50);

3. ஏதேனும் மூன்று DDL கட்டளைகளை எழுதுக.

- CREATE : தரவுத்தளத்தில் அட்டவணைகளை உருவாக்க
- ALTER : தரவுத்தளத்தின் வடிவமைப்பை மாற்றியமைக்க
- DROP : தரவுத்தளத்தில் உள்ள அட்டவணைகளை நீக்க
- TRUNCATE : ஒரு அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் அழிக்கும். மேலும், அவற்றுக்கென ஒதுக்கப்பட்ட இடத்தையும் விடுவிக்கும்.

4. Savepoint கட்டளையின் பயன்பாட்டை ஒரு எடுத்துக்காட்டுடன் எழுதுக.

- SAVEPOINT கட்டளை ஒரு நடவடிக்கையை தற்காலிகமாக சேமித்து வைப்பதால், நாம் தேவைப்படும் சமயங்களில் திரும்பவும் அந்த நிலைக்கு செல்லலாம்.
- ROLLBACK கட்டளையைப் பயன்படுத்தி, மீண்டும் அந்த நிலையை சென்றடையலாம்.

SAVEPOINT savepoint\_name;

- எடுத்துக்காட்டு

| Admno | Name    | Age |
|-------|---------|-----|
| 105   | Sumitha | 21  |
| 106   | Selvam  | 48  |
| 107   | Aakash  | 14  |

UPDATE Student SET Name = 'Mini' WHERE Admno=105;

SAVEPOINT A;

SELECT \* FROM STUDENT;

| Admno | Name        | Age |
|-------|-------------|-----|
| 105   | <i>Mini</i> | 21  |
| 106   | Selvam      | 48  |
| 107   | Aakash      | 14  |

ROLLBACK TO A;

SELECT \* FROM STUDENT;

| Admno | Name           | Age |
|-------|----------------|-----|
| 105   | <i>Sumitha</i> | 21  |
| 106   | Selvam         | 48  |
| 107   | Aakash         | 14  |

5. DISTINCT சிறப்புச் சொல்லை பயன்படுத்தி ஒரு SQL கூற்றினை எழுதுக.

SELECT DISTINCT place FROM student;

பகுதி-ஈ

அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண் )

1. பல்வேறு வகையான கட்டுப்பாடுகளையும், அதன் செயல்பாடுகளையும் எழுதுக.

- கட்டுப்பாடுகள் என்பது ஒரு புலம் அல்லது புலங்களின் தொகுதிக்கு பயன்படுத்தக்கூடிய நிபந்தனையை குறிக்கும்.

**Unique கட்டுப்பாடு**

- குறிப்பிட்ட நெடுவரிசைகளில் எந்த இரு வரிசைகளும் ஒரே மதிப்பை கொண்டிருக்காது என்பதை உறுதி செய்கிறது.
- ஒன்றுக்கும் மேற்பட்ட புலங்களை Unique key ஆக பயன்படுத்தலாம்.
- NOT NULL என்று அறிவிக்கப்பட்ட புலங்களுக்கு மட்டுமே பயன்படுத்த முடியும்.
- எடுத்துக்காட்டு: CREATE TABLE Student

(

Admno integer NOT NULL UNIQUE, → Unique கட்டுப்பாடு

Name char (20) NOT NULL

);

**Primary Key கட்டுப்பாடு**

- தரவுத்தளத்திலுள்ள ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட ஒரு புலத்தினை primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்க வேண்டும்.
- ஒரு புலத்தை மட்டுமே primary key ஆக குறிப்பிட முடியும்.
- வெற்று (NULL) மதிப்புகளை அனுமதிக்காது என்பதால், NOT NULL என்ற கட்டுப்பாட்டையும் கொண்டிருக்கும்.
- எடுத்துக்காட்டு: CREATE TABLE Student

(

Admno integer PRIMARY KEY, → Primary Key கட்டுப்பாடு

Name char(20) NOT NULL

);

**DEFAULT கட்டுப்பாடு**

- ஒரு புலத்தில் கொடாநிலை மதிப்புகளை இருத்தி வைக்க பயன்படுகிறது.
- DEFAULT கட்டுப்பாட்டை கொண்ட குறிப்பிட்ட புலத்திற்கு மதிப்பு கொடுக்காவிடில், தானாகவே கொடாநிலை மதிப்பு அந்த புலத்தில் இருத்தப்பட்டுவிடும்.
- எடுத்துக்காட்டு: CREATE TABLE Student

(

Admno integer PRIMARY KEY,

Name char(20) NOT NULL,

Age integer DEFAULT 17 → Default கட்டுப்பாடு

);

**Check கட்டுப்பாடு**

- இந்தக் கட்டுப்பாடு ஒரு புலத்திற்கான மதிப்பின் வரம்பை நிர்ணயிக்க உதவுகிறது.

- CHECK கட்டுப்பாட்டை ஒரு நெடுவரிசைக்கு வரையறுத்தால், அப்புலத்தில் வரையறுக்கப்பட்ட மதிப்புகளை மட்டுமே உள்ளிட அனுமதிக்கும்.
- எடுத்துக்காட்டு: CREATE TABLE Student  
(  
Admno integer PRIMARY KEY  
Name char(20) NOT NULL,  
Age integer (CHECK Age <=19), → Check கட்டுப்பாடு  
);

### அட்டவணை கட்டுப்பாடு

- ஒரு அட்டவணையில் உள்ள புலங்களின் குழுவிற்கு கட்டுப்பாட்டைப் பயன்படுத்தினால், அதனை அட்டவணைக் கட்டுப்பாடு என்கிறோம்.
- பொதுவாக, அட்டவணை கட்டுப்பாடு அட்டவணை வரையறுப்பின் இறுதியில் குறிப்பிடப்படும்.
- எடுத்துக்காட்டு: CREATE TABLE Student1  
(  
Admno integer NOT NULL,  
Firstname char(20),  
Lastname char(20),  
PRIMARY KEY (Firstname, Lastname) → அட்டவணை கட்டுப்பாடு  
);

### 2. கீழ்க்காணும் பணியாளர் அட்டவணையை கருத்தில் கொண்டு, (i) முதல் (v) வரையிலான வினாக்களுக்கு SQL கட்டளைகளை எழுதுக.

| EMP CODE | NAME      | DESIG      | PAY   | ALLO WANCE |
|----------|-----------|------------|-------|------------|
| S1001    | Hariharan | Supervisor | 29000 | 12000      |
| P1002    | Shaji     | Operator   | 10000 | 5500       |
| P1003    | Prasad    | Operator   | 12000 | 6500       |
| C1004    | Manjima   | Clerk      | 8000  | 4500       |
| M1005    | Ratheesh  | Mechanic   | 20000 | 7000       |

- அனைத்து பணியாளர்களின் விவரங்களை அவர்கள் பெறும் சம்பளங்களின் இறங்கு வரிசையில் காண்பிக்க.
- 5000 முதல் 7000 வரை ALLOWANCE பெறும் அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.
- mechanic வகையை சார்ந்த பணியாளர்களை நீக்க.
- ஒரு புதிய வரிசையை உருவாக்க.
- operators வகையை சார்ந்த அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.

- SELECT \* FROM EMPLOYEE ORDER BY PAY DESC ;
- SELECT \* FROM EMPLOYEE WHERE ALLOWANCE BETWEEN 5000 AND 7000 ;
- DELETE FROM EMPLOYEE WHERE DESIG = 'Mechanic' ;
- INSERT INTO EMPLOYEE VALUES ('S1006', 'Rajesh', 'Mechanic', 20000, 7000) ;
- SELECT \* FROM EMPLOYEE WHERE DESIG = 'Operator' ;

### 3. SQL ன் கூறுகள்? ஒவ்வொன்றிற்கும் கட்டளைகளை எழுதுக.

#### தரவு வரையறை மொழி (DDL)

- தரவுத்தள அமைப்பு அல்லது திட்ட வடிவமைப்பினை வரையறுக்கும் SQL கூற்றுகளை கொண்டிருக்கும்.
  - CREATE – தரவுத்தளத்தில் அட்ட வணை களை உருவாக்க
  - ALTER – தரவுத்தளத்தின் வடிவமைப்பை மாற்றியமைக்க
  - DROP – தரவுத்தளத்தில் உள்ள அட்டவணைகளை நீக்க
  - TRUNCATE – ஒரு அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் அழிக்கும். மேலும், அவற்றுக்கென ஒதுக்கப்பட்ட இடத்தையும் விடுவிக்கும்.

#### தரவு கையாளுதல் மொழி (DML)

- ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை சேர்த்தல், அழித்தல் மற்றும் மாற்றியமைப்பதற்காக பயன்படும் ஒரு வினாவல் மொழியாகும்.
  - INSERT – ஒரு அட்டவணையில் தரவுகளை நுழைத்தல்.

- ii) UPDATE – அட்டவணையில் ஏற்கனவே உள்ள தரவுகளை புதுப்பித்தல்.
- iii) DELETE – அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் நீக்கும். ஆனால் அவற்றிற்கு ஒதுக்கப்பட்ட நினைவக பகுதியை விடுவீக்காது.

#### தரவு கட்டுப்பாடு மொழி (DCL)

- தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளின் அணுகுதலை கட்டுப்படுத்தும் ஒரு நிரலாக்க மொழியாகும்.
  - i) GRANT – ஒரு குறிப்பிட்ட பணியை செய்ய ஒன்று அல்லது அதற்கு மேற்பட்ட பயனர்களுக்கு அனுமதி வழங்கும்.
  - ii) REVOKE – Grant கட்டளையினால் தரப்பட்ட அணுகல் அனுமதி திரும்பப் பெறப்படும்.

#### பரிவர்த்தனை கட்டுப்பாட்டு மொழி (TCL)

- தரவுத்தளத்தில் உள்ள பரிவர்த்தனைகளை நிர்வகிக்க பயன்படுகிறது.
  - i) Commit – தரவுத்தள பரிவர்த்தனையை நிரந்தரமாக சேமிக்கும்.
  - ii) Roll back – ஒரு தரவுத்தளத்தை முந்தைய commit நிலை வரை மீட்டெடுக்கும்.
  - iii) Save point – Rollback செய்வதற்கு ஏதுவாக தரவுத்தள பரிவர்த்தனையை தற்காலமாக சேமிக்கும்.

#### தரவு வினவல் மொழி (DQL)

- தரவு வினவல் மொழி ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை வினவுதலுக்கும், மீட்டெடுப்பதற்கான கட்டளைகளை கொண்டுள்ளது.
  - i) Select – அட்டவணையிலுள்ள பதிவுகளை வெளிக்காட்டும்.

#### 4. மாணவர் அட்டவணையில் பின்வரும் SQL கூற்றுகளை கட்டமைக்கவும்.

- (i) SELECT கூற்று GROUP BY clause பயன்படுத்தி
- (ii) SELECT கூற்று ORDER BY clause பயன்படுத்தி

##### (i) SELECT கூற்று GROUP BY clause

- SELECT கூற்றுடன் GROUP BY clause பயன்படுத்தி வரிசைகள் அல்லது நெடுவரிசைகளில் ஒரே மாதிரியான மதிப்புகளை ஒரு தொகுதியாக பிரிக்கலாம் அல்லது ஒரு அட்டவணையை குழுக்களாக பிரிக்கலாம்.
- தொடரியல்: SELECT <column-names> FROM <table-name> GROUP BY <column-name> HAVING [condition];
- எடுத்துக்காட்டு: SELECT Gender, count(\*) FROM Student GROUP BY Gender;

##### (ii) SELECT கூற்று ORDER BY clause

- SQL-ன் ORDER BY clause பயன்படுத்தி தரவுகளை ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளின் அடிப்படையில் ஏறுவரிசையிலோ அல்லது இறங்கு வரிசையிலோ வரிசைப்படுத்தலாம்.
- தொடரியல்: SELECT <column-name>[,<column-name>,...] FROM <table-name>ORDER BY <column1>, <column2>, ... ASC | DESC ;
- எடுத்துக்காட்டு: SELECT \* FROM Student WHERE Age<18 ORDER BY Name;

#### 5. பணியாளர்களுக்கான ஏதேனும் ஐந்து புலங்களைக் கொண்ட ஒரு அட்டவணையை உருவாக்க ஒரு SQL கூற்றினை எழுதி, அந்த பணியாளர் அட்டவணைக்கு ஒரு அட்டவணை கட்டுப்பாட்டை உருவாக்கவும்.

```
CREATE TABLE Employee
(
 Empcode char(20),
 Name char(20),
 Desig varchar(20),
 Pay integer,
 Allowance integer,
 PRIMARY KEY (Empcode) → அட்டவணை கட்டுப்பாடு
);
```

பாடம்-13 பைத்தான் மற்றும் CSV கோப்புகள்  
பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண் )

1. CSV கோப்பு என்றால் என்ன?

- CSV (Comma Separated values) கோப்பில் ஒவ்வொரு வரியும் காற்புள்ளி அல்லது வேறு பிரிப்பான்களை பயன்படுத்தி பிரிக்கப்பட்ட பல்வேறு புலங்களைக் கொண்டு பயனரால் புரிந்து கொள்ள சூடிய உரைக் கோப்பாகும்.

2. பைத்தான் மூலம் CSV கோப்பை படிப்பதற்கான இரு வழிகளை குறிப்பிடுக.

- csv.reader () செயற்கூறு
- Dict Reader இனக்குழு

3. கோப்பின் கொடாநிலை முறைமைகளை குறிப்பிடுக.

- 'r' என்ற முறைமை படித்தல் முறைமை கோப்பை திறக்கும் கொடாநிலையாகும்.
- 't' என்ற உரை முறைமை படித்தல் மற்றும் எழுதுதல் செயல்பாடுகளில் கொடாநிலையாகும்.

4. next() செயற்கூறின் பயன்பாடு என்ன?

- CSV கோப்பினைத் திறந்து படிக்கும் போது ஒரு வரிசையைத் தவிர்க்க next() செயற்கூறு பயன்படுகிறது.

5. csv கோப்பில் ஒன்றிக்கு மேற்பட்ட நெடுவரிசையை எவ்வாறு வரிசையாக்கம் செய்வாய்? எடுத்துக்காட்டுத் தருக.

- ஒன்றுக்கு மேற்பட்ட புலங்களை வரிசைப்படுத்த itemgetter என்பதில் ஒன்றுக்கு மேற்பட்ட சுட்டு எண்களை குறிப்பிடுவதன் மூலம் நிறைவேற்றலாம்.
- தொடரியல்: operator.itemgetter(col\_no)
- எடுத்துக்காட்டு: sortedlist = sorted (data, key=operator.itemgetter(1))

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண் )

1. பைத்தானிலுள்ள open() செயற்கூற்றை பற்றி குறிப்பு எழுதுக. மேலும் இதன் இரண்டு வழிமுறைகளின் வேறுபாடுகள் என்ன?

- open() என்ற உள்ளிணைந்த செயற்கூறு பைத்தானில் ஒரு கோப்பை திறக்கும்.
- இந்த செயற்கூறு கோப்பு பொருளை திரும்பி கொடுக்கும்,
- இதை பயன்படுத்தி கோப்பின் தரவுகளை படிக்கவோ அல்லது மாற்றவோ முடியும்.

இரண்டு வழிமுறைகள்

```
f = open("sample.txt") :
```

- மேலே குறிப்பிட்டுள்ள வழிமுறை பாதுகாப்பானது அல்ல, ஏதேனும் ஒரு பிழை ஏற்படுமெனில் அந்த செயற்பாட்டிலிருந்து நிரலானது கோப்பை மூடாமல் வெளியேறும்.

```
with open("sample.txt",'r') as f :
```

- with தொகுதியிலிருந்து வெளியேறும் போது அதனுள்ளே கொடுக்கப்பட்டுள்ள கோப்பு மூடப்படும். close() செயற்கூற்றை வெளிப்படையாக கொடுக்க வேண்டியதில்லை.

2. ஏற்கனவே உள்ள கோப்பில் மாற்றம் செய்யும். பைத்தான் நிரலை எழுதுக.

```
import csv
row = ['3', 'Meena','Bangalore']
with open('student.csv', 'r') as readFile:
 reader = csv.reader(readFile)
 lines = list(reader)
 lines[3] = row
with open('student.csv', 'w') as writeFile:
 writer = csv.writer(writeFile)
 writer.writerows(lines)
readFile.close()
writeFile.close()
```

3. காற்புள்ளியை (,) தானமைவு பிரிப்பானாக கொண்டுள்ள CSV கோப்பினை படிப்பதற்கான பைத்தான் நிரலை எழுதுக.

```
import csv
```

```

with open('student.csv','r') as readfile:
 reader=csv.reader(readfile)
 for row in reader:
 print(row)
readfile.close()

```

#### 4. write மற்றும் append mode முறைமைகளின் வேறுபாடு என்ன?

|                                                                                                              |                                                                                   |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| write முறைமை ( 'w' )                                                                                         | append முறைமை ( 'a' )                                                             |
| கோப்பில் தரவுகளை எழுதுவதற்கு திறக்கவும்.                                                                     | கோப்பின் தரவுகளை அழிக்காமல் அதன் இறுதியில் புதிய தரவுகளை சேர்ப்பதற்கு திறக்கவும். |
| குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும். கோப்பில் தரவுகள் இருப்பின் அவை அழிக்கப்படும். | குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும்.                    |

#### 5. reader() மற்றும் DictReader() இடையேயான வேறுபாடு யாது?

|                                                                                                     |                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| reader()                                                                                            | DictReader()                                                                                                                          |
| reader() செயற்கூறானது கோப்பின் ஒவ்வொரு வரியையும் படித்து அவற்றை நெடுவரிசைகளின் பட்டியலாக அமைக்கும். | DictReader இனக்குழு CSV கோப்பில் உள்ள நெடுவரிசையானது Dictionary-யின் மதிப்புகளாக செயல்பட்டு அவற்றை உரிய திறவுகோள் மூலம் அணுக முடியும் |
| csv.reader மற்றும் csv.writer ஆனது பட்டியல் (list/tuple) பதிவுடன் வேலை செய்யும்.                    | DictReader மற்றும் csv.DictWriter ஆனது அகராதியில் வேலை செய்யும்.                                                                      |

#### பகுதி-II

#### அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண் )

#### 1. எக்ஸெல் மற்றும் CSV கோப்பின் வேறுபாடு என்ன?

| எக்ஸெல்                                                                                                                      | CSV                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| கோப்பின் அனைத்து அட்டவணைத்தாளிலுள்ள பொருளடக்கம் மற்றும் வடிவாக்கங்களை உள்ளடக்கிய தகவல்களை ஒரு நிலை வடிவில் கொண்ட கோப்பாகும். | CSV வடிவத்தில் காற்புள்ளிகளால் பிரிக்கப்பட்ட தொடர்ச்சியான மதிப்புகளைக் கொண்ட எளிய உரை வடிவ கோப்பாகும்.                 |
| XLS கோப்புகள் அவற்றை படிப்பதற்காக மற்றும் எழுதுவதற்காக அவற்றை உருவாக்கப்பட்ட பயன்பாடுகளை கொண்டு மட்டுமே திறக்கலாம்.          | CSV கோப்புகளை Windows இயக்க அமைப்பில் உள்ள notepad, MS Excel, OpenOffice போன்ற உரைப் பதிப்பான்களைக் கொண்டு திறக்கலாம். |
| அதற்கென உருவாக்கப்பட்டுள்ள தனியுரிம வடிவமைப்பு வகையில் கோப்பினை சேமிக்கும். (xls அல்லது xlsx).                               | அட்டவணை வடிவ தகவல்களை .csv என்ற நீட்டிப்புடன் பிரிக்கப்பட்ட உரைக் கோப்புகளாக சேமிக்கும் வடிவம் CSV ஆகும்.              |
| தரவை பெறும் இறக்கம் செய்யும் (import) போது அதிக நினைவக இடத்தை எடுத்துக் கொள்ளும்.                                            | கோப்புகளை இறக்கம் செய்யும் போது வேகமாக செயல்படும். குறைவான நினைவக இடத்தை எடுத்துக் கொள்ளும்.                           |

#### 2. பல்வேறு கோப்பு முறைமைகளின் பொருள்களை பட்டியலிடுக.

| முறை | விளக்கம்                                                                                                                                              |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 'r'  | படிப்பதற்கு மட்டுமே ஒரு கோப்பினை திறக்கவும் (தானமைவு நிலை)                                                                                            |
| 'w'  | கோப்பில் தரவுகளை எழுதுவதற்கு திறக்கவும். குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும். கோப்பில் தரவுகள் இருப்பின் அவை அழிக்கப்படும். |
| 'x'  | தனித்துவமான படைப்பிற்காக கோப்பினை திறக்கவும். கோப்பு முன்பே உருவாக்கப்பட்டிருந்தால் இந்த செயல் முறையானது தோல்வியடையும்.                               |
| 'a'  | கோப்பின் தரவுகளை அழிக்காமல் அதன் இறுதியில் புதிய தரவுகளை சேர்ப்பதற்கு திறக்கவும். குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும்.      |
| 't'  | உரை முறைமையில் கோப்பு திறக்கவும் (தானமைவு நிலை )                                                                                                      |
| 'b'  | இருமநிலை முறைமையில் கோப்பினை திறக்கவும்.                                                                                                              |
| '+'  | புதுப்பித்தலிற்காக கோப்பினை திறக்கவும் (படித்தல் மற்றும் எழுதுதல்)                                                                                    |

### 3. பைத்தானில் ஒரு கோப்பை படிப்பதற்கான பல்வேறு வழிமுறைகளை எழுதுக.

- கோப்பினை படிக்க இராண்டு வழி முறைகள் உள்ளன.

i. CSV தொகுதி reader() செயற்கூறை பயன்படுத்துதல்

ii. DictReader இனக்குழுவை பயன்படுத்துதல்

#### CSV-ன் reader() செயற்கூறு:

- reader() செயற்கூறானது கோப்பின் ஒவ்வொரு வரியையும் படித்து அவற்றை நெடுவரிசைகளின் பட்டியலாக அமைக்கும்.
- இச்செயற்கூறை பயன்படுத்தி பயனர் கோப்பின் தரவுகளில் உள்ள இரட்டை மேற்கோள் குறி (" " ) , ( | ) மற்றும் ( , ) போன்ற பல்வேறு வடிவமைப்புகளை பயன்படுத்தி படிக்கலாம்.
- csv.reader மற்றும் csv.writer ஆனது பட்டியல் (list/tuple) பதிவுடன் வேலை செய்யும்.
- தொடரியல் : csv.reader(fileobject, delimiter, fmtparams)
- இங்கு,  
file object : கோப்பின் பாதையையும் முறைமையும் திருப்பி அனுப்பும்.  
delimiter : இது செந்தர வரம்புக் குறிகளை கொண்டிருக்கும். (',' ) மற்றவைகள் தவிர்க்கப்படும். இது விருப்பத்தேர்வு அளபுருவாகும்.  
fmtparams : கொடாநிலை மதிப்புகளை மேற்பதிப்பு செய்ய / நீக்க பயன்படும்.  
இது விருப்பத்தேர்வு அளபுருவாகும்.

- எடுத்துக்காட்டு:

```
import csv
with open('c:\pyprg\sample1.csv', 'r') as F:
 reader = csv.reader(F)
 for row in reader :
 print(row)
F.close()
```

#### DictReader இனக்குழு

- DictReader இனக்குழு CSV கோப்பில் உள்ள நெடுவரிசையானது Dictionary-யின் மதிப்புகளாக செயல்பட்டு அவற்றை உரிய திறவுகோள் மூலம் அணுக முடியும்.
- DictReader மற்றும் csv.DictWriter ஆனது அகராதியில் வேலை செய்யும்.
- எடுத்துக்காட்டு:

```
import csv
filename="Student.csv"
input_file=csv.DictReader(open(filename,'r'))
for row in input_file:
 print(dict(row))
```

### 4. தனிப்பயனாக்கம் பிரிப்பானுடன் கூடிய CSV கோப்பை எழுதுவதற்கான பைத்தான் நிரலை எழுதுக.

```
import csv
csv.register_dialect('myDialect', delimiter = '|', quoting=csv.QUOTE_ALL)
with open('grade.csv', 'w') as csvfile:
 fieldnames = ['Name', 'Grade']
 writer = csv.DictWriter(csvfile, fieldnames=fieldnames, dialect="myDialect")
 writer.writeheader()
 writer.writerows ([{'Grade': 'B', 'Name': 'Anu'},
 {'Grade': 'A', 'Name': 'Beena'},
 {'Grade': 'C', 'Name': 'Tarun'}])
 print("writing completed")
```

### 5. CSV கோப்பிலுள்ள தரவை வடிவமைப்பதற்கு பின்பற்ற வேண்டிய விதிமுறைகளை எழுதுக.

1. ஒவ்வொரு வரிசையும் புதிய வரியில் இருந்த அந்த வரியை நுழைவு பொத்தானை அழுத்த வேண்டும். எடுத்துக்காட்டு :

xxx,yyy ←

← என்ற குறியீடானது நுழைவு விசையை குறிக்கும்.

2. கோப்பினில் உள்ள கடைசி பதிவானது வரிமுறிவு/வரி செலுத்தி பிரிப்பானைக் கொண்டிருக்கலாம் அல்லது இல்லாமலும் இருக்கலாம்.எடுத்துக்காட்டாக:

```
ppp,qqq ←
yyy,xxx
```

3. சாதாரண பதிவு வரிசைகளின் வடிவங்களை போன்று கோப்பின் முதல் வரிசையில் தோன்றக்கூடிய விருப்பத்தலைப்பு வரிசை இருக்கலாம். கோப்பின் தலைப்பானது புலங்களின் தொடர்புடைய பெயரினை கொண்டிருக்க வேண்டும். மேலும் பதிவுகளிலுள்ள புலங்களின் எண்ணிக்கையில் மீதமுள்ள கோப்பில் இருத்தல் வேண்டும்.

எடுத்துக்காட்டு

```
field_name1, field_name2, field_name3 ←
aaa,bbb,ccc ←
zzz,yyy,xxx CRLF(Carriage Return and Line Feed)
```

4. தலைப்பு மற்றும் ஒவ்வொரு பதிவிலும் ஒன்று அல்லது அதற்கு மேற்பட்ட புலங்கள் காற்புள்ளியால் பிரிக்கப்பட்டிருக்கலாம். இடைவெளியானது புலத்தின் ஒரு பகுதியான கருதப்படும் மேலும் நிராகரிக்கப்பட மாட்டாது. பதிவின் கடைசி காற்புலத்தை தொடர்ந்து காற்புள்ளி இடம் பெறல் கூடாது.

எடுத்துக்காட்டு: Red, Blue

5. ஒவ்வொரு புலமும் இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கப்படலாம் அல்லது கொடுக்கப்படாமல் இருக்கலாம். புலமானது இரட்டை மேற்கோள் குறிகளுக்குள் தரப்படவில்லையெனில், புலங்களில் இரட்டை மேற்கோள் குறியானது தோன்றாது.

எடுத்துக்காட்டு:

```
"Red","Blue","Green" ← # இரட்டை மேற்கோள் குறிகளுடன் உள்ள புலத்தின் தரவு
Black,White,Yellow # இரட்டை மேற்கோள் குறிகள் இல்லாத புலத்தின் தரவு
```

6. புலங்களில் வரிதிருப்பி (CRLF), இரட்டை மேற்கோள் குறி மற்றும் காற்புள்ளி போன்றவைகளை கொண்டிருந்தால் அவைகள் இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கப்பட வேண்டும்.

எடுத்துக்காட்டு:

```
Red, ",", Blue CRLF # காற்புள்ளியே ஒரு புலத்தின் மதிப்பாக இருப்பதால் அதை இரட்டை
மேற்கோள் குறிக்குள் கொடுக்கப்பட வேண்டும்.
Red, Blue , Green
```

7. புலமானது இரட்டை மேற்கோள் குறிக்களுக்குள் கொடுக்கப்பட்டிருக்க வேண்டுமெனில், இரட்டை மேற்கோள் குறிகளுக்குள் உள்ள புலமானது மற்றொரு இரட்டை மேற்கோள் குறிகளுக்குள் அமைக்கப்பட வேண்டும்.

```
" "Red", "Blue", "Green" CRLF # இரட்டை மேற்கோள்குறி ஒரு புலத்தின் மதிப்பாக , ,
White இருப்பதால் அதை மற்றொரு இரட்டை மேற்கோள் குறிக்குள் அமைக்கப்பட
வேண்டும்
```

## பாடம்-14 பைத்தானில் C++ நிரல்களை தருவித்தல்

### பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண் )

1. Scripting மொழிக்கும், மற்ற நிரலாக்க மொழிக்கும் உள்ள தத்துவர்த்த வேறுபாடு யாது?

| Scripting மொழி                                                                  | நிரலாக்க மொழி                                                         |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Scripting மொழிக்கு தொகுத்தல் படிநிலைத் தேவைப்படாது, மாறாக விளக்கம் தேவைப்படும். | நிரலாக்க மொழிக்கு நிரல் இயக்குவதற்கும் முன்பாக, தொகுக்கப்பட வேண்டும். |
| Scripting மொழிக்கு வரி மொழி மாற்றித் தேவைப்படுகிறது.                            | நிரலாக்க மொழிக்கு தொகுப்பான் / நிரல் பெயர்ப்பி தேவைப்படுகிறது.        |
| எடுத்துக்காட்டு: PHP, Perl, Python                                              | எடுத்துக்காட்டு: C, C++, JAVA                                         |

2. தொகுப்பான் மற்றும் வரிமொழி மாற்றியை வேறுபடுத்துக.

| தொகுப்பான்                                                                | வரிமொழி மாற்றி                                         |
|---------------------------------------------------------------------------|--------------------------------------------------------|
| முழு நிரலையும் ஸ்கேன் செய்த பிறகு நிரலை இயந்திர மொழியாக மொழி பெயர்க்கும். | ஒவ்வொரு வரியாக நிரலை இயந்திர மொழியாக மொழி பெயர்க்கும். |
| அனைத்து பிழைகளும் மொத்தமாக காட்டும்.                                      | முதல் பிழை ஏற்படும் வரை மொழிபெயர்ப்பி தொடரும்.         |
| பிழைகளை திருத்தம் செய்வது கடினம்.                                         | பிழைகளை திருத்தம் செய்வது எளிது.                       |

3. விரிவாக்கம் தருக (i) SWIG (ii) MinGW

- i) SWIG – Simplified Wrapper Interface Generator
- ii) MinGW – Minimalist GNU for Windows

4. கூறுநிலைகளின் பயன் யாது?

- கூறுநிலை நிரலாக்கம் என்பது குறிமுறையை சிறுசிறு பகுதிகளாக பிரிப்பதற்கான மென்பொருள் வடிவமைப்பு நுட்பமாகும்.
- சார்பு நிலையை குறைக்கிறது.
- குறிமுறையின் மறுபயனாக்கத்தை வழங்குகிறது.

5. cd கட்டளையின் பயன் யாது? எடுத்துக்காட்டு தருக.

- தொடரியல்: cd <absolute path>
  - இதில் cd கட்டளை change directory என்பதையும்.
  - absolute path என்பது செல்ல வேண்டிய கோப்புறையின் முழுமையான பாதையையும் குறிக்கும்.
- எடுத்துக்காட்டு: C:\>cd e:\myfiles

பகுதி-இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண் )

1. பைத்தான் மற்றும் C++ வேறுபடுத்துக

| பைத்தான்                                                  | C++                                                   |
|-----------------------------------------------------------|-------------------------------------------------------|
| 'வரி மொழி மாற்றி' மொழியாகும்                              | நிரல்பெயர்ப்பி மொழியாகும்                             |
| மாறும் தன்மைக் கொண்ட மொழியாகும்                           | நிலையான தன்மைக் கொண்ட தொகுக்கப்பட்ட மொழி              |
| மாறியை அறிவிக்கும் போது, தரவினத்தை குறிப்பிடத் தேவையில்லை | மாறியை அறிவிக்கும் போது, தரவினத்தை குறிப்பிட வேண்டும் |
| Scripting மற்றும் பொதுப்பயன் மொழியாகும்                   | பொதுப்பயன் நிரலாக்க மொழியாகும்                        |

2. Scripting மொழியின் பயன்பாடுகள் யாவை?

- ஒரு நிரலில் சில செயல்பாடுகளை தானியங்குப்படுத்துதல்.
- தரவு தொகுப்பிலிருந்து தகவலைப் பிரித்தெடுத்தல்.
- பழமையான நிரலாக்க மொழிகளுடன் ஒப்பிடும் போது, குறைந்த நிரல் குறிமுறையைக் கொண்டது.
- பயன்பாடுகளுக்கு புதிய செயல்பாடுகளை கொண்டு வர முடியும்.
- சிக்கலான அமைப்புகளை ஒருங்கமைக்க முடியும்.

3. MinGW என்றால் என்ன? அதன் பயன் யாது?

- MinGW ஒரு இடைமுகமாகும்.
- MinGW இயக்க நேர தலைப்புக் கோப்புகளின் தொகுப்பைக் குறிக்கிறது.
- MinGW-W64 windows இன் C++ க்கு சிறந்த தொகுப்பான் ஆகும்.
- MinGW, g++-ஐ பயன்படுத்தி பைத்தான் நிரல் மூலம் C++ நிரல்களை தொகுத்து, இயக்க அனுமதிக்கிறது.

4. கீழ்க்காணும் கூற்றில் கூறுநிலை, செயற்குறி, வரையறையின் பெயர் ஆகியவற்றை அடையாளம் காண்க. welcome.display()

- welcome → கூறுநிலையின் பெயர்
- . → புள்ளி செயற்குறி
- display( ) → வரையறையின் பெயர்

## 5. sys.argv என்றால் என்ன?

- sys.argv என்பது பைத்தான் நிரலுக்கு அனுப்பி வைக்கப்படும் கட்டளை வரி செயலுருபுகளின் பட்டியலாகும்.
- argv கட்டளை வரி உள்ளீட்டு வழியாக வரும் உருப்படிகள் அனைத்தையும் கொண்டிருக்கும்.
- sys.argvஐ பயன்படுத்த, முதலில் நீங்கள் sys கூறுநிலையைத் தருவித்துக் கொள்ள வேண்டும்.
- முதல் செயலுருபு sys.argv[0] எப்பொழுதும் செயல்படுத்த வேண்டிய நிரலின் பெயராக இருக்கும்.
- மேலும், sys.argv[1] என்பது நிரலுக்கு அனுப்பப்படும் முதல் செயலுருபு ஆகும்.

### பகுதி-ஈ

## அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண் )

### 1. பைத்தானில் ஏதேனும் 5 பண்புக்கூறுகளை கூறவும்.

- பைத்தான் பயனிலா (garbage) மதிப்புகளைச் சேகரிக்கும் தானியங்கியைப் பயன்படுத்துகிறது.
- பைத்தான் ஒரு மாறக்கூடிய வகையைச் சார்ந்த மொழியாகும்.
- பைத்தான் வரி மொழி மாற்றி மூலம் இயங்குகிறது.
- C++ நிரல் குறிமுறையைக் காட்டிலும் பைத்தான் குறிமுறை 5 - 10 மடங்கு குறைவானது.
- பைத்தானில், வெளிப்படையாக தரவினங்களை அறிவிக்க தேவையில்லை.
- பைத்தானில், ஒரு செயற்கூறு எந்த வகை செயலுருபையும் ஏற்கும்.
- மேலும், முன்னதாக எந்த ஒரு அறிவிப்பும் இல்லாமல் பல மதிப்புகளை திருப்பியனுப்பும்.

### 2. பின்வரும் கட்டளை ஒவ்வொன்றையும் விளக்கவும்.

Python <filename.py> -<i> <C++ filename without cpp extension>

|                                    |                                                                        |
|------------------------------------|------------------------------------------------------------------------|
| Python                             | கட்டளை வரியிலிருந்து பைத்தான் நிரலை செயல்படுத்துவதற்கான சிறப்புச் சொல் |
| filename.py                        | செயல்படுத்த வேண்டியபைத்தான் நிரலின் பெயர்                              |
| - i                                | உள்ளீட்டு முறைமை                                                       |
| C++ filename without cpp extension | தொகுக்கப்பட்டு செயல்படுத்தப்பட வேண்டிய C++ நிரலின் பெயர்.              |

### 3. பைத்தானில், sys.os.getopt கூறுநிலைகளின் தேவை என்ன என்பதை விளக்குக.

#### பைத்தானின் sys கூறுநிலை

- sys.argv என்பது பைத்தான் நிரலுக்கு அனுப்பி வைக்கப்படும் கட்டளை வரி செயலுருபுகளின் பட்டியலாகும்.
- argv கட்டளை வரி உள்ளீட்டு வழியாக வரும் உருப்படிகள் அனைத்தையும் கொண்டிருக்கும்.
- sys.argvஐ பயன்படுத்த, முதலில் நீங்கள் sys கூறுநிலையைத் தருவித்துக் கொள்ள வேண்டும்.
- முதல் செயலுருபு sys.argv[0] எப்பொழுதும் செயல்படுத்த வேண்டிய நிரலின் பெயராக இருக்கும்.
- மேலும், sys.argv[1] என்பது நிரலுக்கு அனுப்பப்படும் முதல் செயலுருபு ஆகும்.

#### பைத்தானில் os கூறுநிலை

- பைத்தானில் இருக்கும் OS கூறுநிலை இயக்க அமைப்பு சார்பு செயல்பாட்டுடன் பயன்படுத்துவதற்கான ஒரு வழிமுறையை வழங்குகிறது.
- பைத்தான் இயங்கிக் கொண்டிருக்கும் போது, விண்டோஸ் இயக்க முறைமையுடன் OS கூறுநிலை ஊடாட அனுமதிக்கும் செயற்கூறுகளாவன.
  - **os.system():** செயல்தளத்தில்(Shell)C++ தொகுத்தலுக்கான கட்டளையை இயக்கும்.
  - **கட்டளை:** os.system ('g++' + <variable\_name1> '-<mode>' + <variable\_name2>)
  - **எடுத்துக்காட்டு:** os.system('g++ ' + cpp\_file + ' -o ' + exe\_file)

#### பைத்தானின் getopt கூறுநிலை

- பைத்தானில் getopt கூறுநிலை கட்டளை வரி தேர்வுகளையும், செயலுருப்புகளையும் பிரித்தெடுக்க செயற்கூறுகளை வழங்குகிறது.
- **getopt.getopt செயற்கூறு**
  - இந்த செயற்கூறு கட்டளை வரி தேர்வுகளையும், அளபுருக்களின் பட்டியலையும் பிரித்தெடுக்கும்.
  - **தொடரியல்:** <opts>,<args>=getopt.getopt(argv, options, [long\_options])

4. getopt() என்ற செயற்கூறின் தொடரியலை எழுதி, அதன் செயலுருபுகளையும், திருப்பியனுப்பும் மதிப்புகளையும் விளக்குக.

getopt.getopt செயற்கூறு

- இந்த செயற்கூறு கட்டளை வரி தேர்வுகளையும், அளபுருக்களின் பட்டியலையும் பிரித்தெடுக்கும்.
- தொடரியல் <opts>,<args>=getopt.getopt(argv, options, [long\_options])

அளபுருக்களின் விவரங்களை இங்கே காணலாம்,

- argv → இது பிரிக்கப்பட வேண்டிய அளபுருக்களின் மதிப்புகளின் பட்டியலைக் குறிக்கும்.
- options → இது உள்ளீட்டு அல்லது வெளியீட்டு முறைகளை குறிக்கும் சர எழுத்து தேர்வுகளாகும்.
- long\_options → இந்த அளபுரு சரங்களின் பட்டியலை கொண்டுள்ளது. Long options—ன் செயலுருபைத் தொடர்ந்து '=' சமக்குறி இடம்பெற வேண்டும். நம்முடைய நிரலில் C++ கோப்பின் பெயர், அதனுடன் 'i' என்ற தேர்வும் செலுத்தப்படும்.

getopt வழிமுறை இரண்டு உறுப்புகளை கொண்டுள்ள மதிப்புகளை திருப்பியனுப்பும்.

- Opts → முறைமை, பாதைப் போன்ற பிரிக்கப்பட்ட சரங்களின் பட்டியலைக் கொண்டிருக்கும்.
- Args → தவறான பாதை அல்லது முறைமையின் காரணமாக பிரிக்கப்பட முடியாத எந்தவொரு சரத்தின் பட்டியலைக் கொண்டிருக்கும்.
- getopt() வழிமுறையின் மூலம் சரங்களை பிரித்தெடுக்கும் போது பிழையேதும் இல்லாவிட்டால் args வெற்று அணியாக அமையும்.

5. கீழ்க்காணும் c++ நிரலை செயல்படுத்த ஒரு பைத்தான் நிரலை எழுதவும்.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ cout<<"WELCOME"; return(0); } The above C++ program is saved in a file welcome.cpp
```

```
import sys, os, getopt
def main(argv):
 opts, args = getopt.getopt(argv, "i:")
 for o, a in opts:
 if o in "-i":
 run(a)
def run(a):
 inp_file=a+'.cpp'
 exe_file=a+'.exe'
 os.system('g++ ' + inp_file + ' -o ' + exe_file)
 os.system(exe_file)
if __name__=='__main__':
 main(sys.argv[1:])
```

பாடம்-15 SQL மூலம் தரவுகளைக் கையாளுதல்

பகுதி-ஆ

அனைத்து வினாக்களுக்கும் விடையளி

(2 மதிப்பெண்)

1. தரவுத்தளத்தைப் பயன்படுத்தும் பயனர்களை குறிப்பிடவும்.

- தரவுத்தளத்தின் பயனர்களாக மனிதர்கள், பிற நிரல்கள் அல்லது பயன்பாடுகள் இருக்கலாம்.

2. தரவுத்தளத்தை இணைக்க பயன்படும் முறைகள் யாவை? எடுத்துக்காட்டு தருக.

- connect( ) வழிமுறையைப் பயன்படுத்தி இணைப்பை உருவாக்கி தரவுத்தளத்தின் பெயரை இணைக்கவும்.
- இல்லாவிடில், பைத்தான், கொடுக்கப்பட்ட பெயரில் ஒரு புதிய தரவுத்தளத்தை உருவாக்கும்.
- எடுத்துக்காட்டு:

```
import sqlite3
connection = sqlite3.connect ("Academy.db")
cursor = connection.cursor()
```

### 3. புலத்தை "INTEGER PRIMARY KEY" என அறிவிப்பதன் நன்மை என்ன ?

- அட்டவணையில் உள்ள ஒரு நெடுவரிசை INTEGER PRIMARY KEY, என்று அறிவிக்கப்பட்டால்,
  - NULL என்ற மதிப்பு உள்ளீடாகப் பயன்படுத்தப்படுகிறதோ, அந்த NULL மதிப்பு தானாகவே அந்த நெடுவரிசையில் இது வரை பயன்படுத்தப்பட்ட மிக உயர்ந்த மதிப்பை விட ஒன்று மிகுந்து முழு எண்ணாக இருக்கும்.
  - வெற்று அட்டவணை எனில் 1 என்ற மதிப்பு பயன்படுத்தப்படும்.

### 4. அட்டவணையில் பதிவுகளை விரிவுப்படுத்துவதற்கான கட்டளையை எழுதுக. எ.கா. தருக.

- "INSERT" கட்டளையை SQLiteல் அனுப்புவதன் மூலம் அட்டவணையில் தரவுகளை உள்ளிடலாம். execute() செயற்கூறு கொடுக்கப்பட்ட SQL கட்டளையை செயல்படுத்தும்.
- எடுத்துக்காட்டு: cursor.execute("INSERT INTO Student (Rollno, Sname) VALUES (1561, 'Aravind');")

### 5. தரவுத்தள அட்டவணையிலிருந்து அனைத்து பதிவுகளையும் பெறுவதற்கான வழிமுறை எது?

- fetchall () செயற்கூறு அனைத்து வரிசைகளையும் தரவுத்தள அட்டவணையில் இருந்து பெற பயன்படுகிறது.
- எடுத்துக்காட்டு: 

```
result = cursor.fetchall()
print(result)
```

### பகுதி-இ

#### அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண்)

#### 1. SQLite என்றால் என்ன? இதன் நன்மைகள் யாவை?

- SQLite என்பது எளிய உறவுநிலை தரவுத்தள அமைப்பாகும்.

#### நன்மைகள்

- இது MySQL மற்றும் Oracle போன்ற பயன்பாடுகளுடன் ஒன்றிணைந்து செயல்படும் வகையில் வடிவமைக்கப்பட்டுள்ளது
- வேகமாகவும், மிகுந்த சோதிக்கப்பட்டதாகவும் மற்றும் நெகிழ்வானதாகவும் உள்ளதால் SQLite-ல் வேலை செய்வது எளிதாகும்.

#### 2. fetchone() மற்றும் fetchmany() வேறுபடுத்துக.

| fetchone()                                                                                                                                                  | fetchmany()                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| fetchone() செயற்கூறு வினாவல் முடிவுத் தொகுதியின் உள்ளே உள்ள அடுத்த வரிசையைக் கொடுக்கும் (அல்லது) எந்த வரிசையும் இல்லை என்றால் None என்ற மதிப்பை கொடுக்கும். | குறிப்பிட்ட எண்ணிக்கையிலான பதிவுகளைக் காண்பிக்க fetchmany(n) செயற்கூறு பயன்படுகிறது. |
| while மடக்கு மற்றும் fetchone() செயற்கூறு பயன்படுத்தி அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் காண முடியும்.                                                   | இந்த செயற்கூறு முடிவுத் தொகுதியில் கொடுக்கப்பட்ட எண்ணிக்கையில் பதிவுகளை திருப்பும்.  |

#### 3. Where துணை நிலைக்கூற்றின் பயன் என்ன? where கூற்றைப் பயன்படுத்தி ஒரு பைத்தான் கூற்றை எழுதவும்.

- குறிப்பிட்ட நிபந்தனைகளுக்கு உட்பட்ட பதிவுகளை மட்டுமே பிரித்தெடுக்க WHERE துணை நிலை கூற்று பயன்படுகிறது.
- எடுத்துக்காட்டு: 

```
cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")
```

#### 4. பின்வரும் விவரங்களை படிக்கவும். அதன் அடிப்படையில் துறை வாரியாக பதிவுகளை திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

தரவுத்தள பெயர்: organization.db அட்டவணை பெயர்: Employee புலங்கள்: Eno, EmpName, Esal, Dept

```
import sqlite3
connection=sqlite3.connect("organization.db")
cursor=connection.cursor()
cursor.execute("create table employee(eno integer primary key, empname varchar(20),
esal integer, dept varchar(20));")
cursor.execute("insert into employee values ('1001', 'Raja', '10500', 'Sales');")
cursor.execute("insert into employee values ('1002', 'Surya', '9000', 'Purchase');")
cursor.execute("insert into employee values ('1003', 'Peter', '8000', 'Production');")
```

```

connection.commit()
print("Employee Details Department-wise :")
cursor.execute("Select * from employee order by dept;")
result=cursor.fetchall()
print(*result, sep="\n")
connection.close()

```

5. பின்வரும் விவரங்களை படிக்கவும் அதன் அடிப்படையில் பதிவுகளை Eno இறங்குவரிசையில் திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

```

தரவுத்தள பெயர்: organization.db அட்டவணை பெயர்: Employee புலங்கள்: Eno, EmpName, Esal, Dept
import sqlite3
connection=sqlite3.connect("organization.db")
cursor=connection.cursor()
cursor.execute("create table employee(eno integer primary key, empname varchar(20),
 esal integer, dept varchar(20));")
cursor.execute("insert into employee values ('1001', 'Raja', '10500','Sales');")
cursor.execute("insert into employee values ('1002', 'Surya', '9000','Purchase');")
cursor.execute("insert into employee values ('1003', 'Peter', '8000','Production');")
connection.commit()
print("Records in descending order of employee number:")
cursor.execute("Select * from employee order by eno desc")
result=cursor.fetchall()
print(*result,sep="\n")
connection.close()

```

### பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண்)

1. SQLite பற்றி விரிவாக எழுதவும். அதனை பயன்படுத்தும் படிநிலைகளை எழுதுக.

- SQLite என்பது எளிய உறவுநிலை தரவுத்தள அமைப்பாகும்.
- இது MySQL மற்றும் Oracle போன்ற பயன்பாடுகளுடன் ஒன்றிணைந்து செயல்படும் வகையில் வடிவமைக்கப்பட்டுள்ளது
- வேகமாகவும், மிகுந்த சோதிக்கப்பட்டதாகவும் மற்றும் நெகிழ்வானதாகவும் உள்ளதால் SQLite-ல் வேலை செய்வது எளிதாகும்.
- SQLite ற்காக பைத்தான் உள்ளிணைந்த நூலகத்தைக் கொண்டுள்ளது.

பயன்படுத்தும் படிநிலைகள்:

- **படிநிலை 1 :**
  - sqlite3யை இணைக்கவும்
- **படிநிலை 2 :**
  - connect( ) வழிமுறையைப் பயன்படுத்தி இணைப்பை உருவாக்கி தரவுத்தளத்தின் பெயரை இணைக்கவும்.
  - அவ்வாறு அனுப்பும்போது, அத்தரவுத்தளம் ஏற்கனவே இருக்குமாயின் அது இணைக்கப்படும். இல்லாவிடில், பைத்தான், கொடுக்கப்பட்ட பெயரில் ஒரு புதிய தரவுத்தளத்தை உருவாக்கும்.
- **படிநிலை 3 :**
  - cursor= connection.cursor( ) என்றக் கூற்றைப் பயன்படுத்தி cursor என்னும் பொருளை அனுப்பவும்.
  - cursor என்பது ஒரு கட்டுப்பாட்டு அமைப்பாகும், இது தரவுத்தளப் பதிவுகளை இணைக்கப் பயன்படுகிறது. பைத்தானில், cursor மிக முக்கிய பங்குவகின்றது.
  - அனைத்து கட்டளைகளும் cursor பொருள் மூலம் மட்டுமே இயக்கப்படும்.
  - எடுத்துக்காட்டு: sql\_comm = "SQL statement"

2. fetchmany() பயன்படுத்தி பின்வரும் அட்டவணையிலுள்ள அனைத்து பதிவுகளையும் திரையிடுவதற்கான பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்

| Icode | ItemName | Rate  |
|-------|----------|-------|
| 1003  | Scanner  | 10500 |
| 1004  | Speaker  | 3000  |
| 1005  | Printer  | 8000  |
| 1008  | Monitor  | 15000 |
| 1010  | Mouse    | 700   |

```
import sqlite3
connection=sqlite3.connect("inventory.db")
cursor=connection.cursor()
cursor.execute("select * from product;")
print("Displaying all records in the table")
result=cursor.fetchmany(5)
print(*result, sep="\n")
connection.close()
```

### 3. HAVING துணை நிலைக்கூற்றின் பயன் யாது? எடுத்துக்காட்டு தருக.

- குழு சார்புகளைப் பொறுத்து தரவுகளை வடிகட்ட HAVING துணை நிலைக்கூற்று பயன்படுகிறது.
- இது WHERE நிபந்தனை கூற்றை ஒத்ததாகும் ஆனால் குழு சார்புகளுடன் பயன்படுகிறது.
- குழு சார்புகளை WHERE துணை நிலைக் கூற்றில் பயன்படுத்த முடியாது. ஆனால், HAVING துணை நிலைக் கூற்றில் பயன்படுத்த முடியும்.

- எடுத்துக்காட்டு
 

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student
GROUP BY GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

- வெளியீடு ['gender', 'COUNT(GENDER)']  
[('M', 5)]

### 4. பின்வரும் குறிப்புகளைக் கொண்டு ITEM என்ற அட்டவணையை உருவாக்க பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

அட்டவணைக்கு ஒரு பதிவை சேர்க்கவும்.

தரவுத்தளத்தின் பெயர் :- ABC

அட்டவணையின் பெயர் :- Item

நெடுவரிசையின் பெயர் மற்றும் விவரங்கள் :-

|                    |    |                                |
|--------------------|----|--------------------------------|
| Icode              | :- | integer and act as primary key |
| Item Name          | :- | Character with length 25       |
| Rate               | :- | Integer                        |
| Record to be added | :- | 1008, Monitor,15000            |

```
import sqlite3
connection=sqlite3.connect("ABC.db")
cursor=connection.cursor()
cursor.execute("drop table item;")
cursor.execute("create table item(icode integer primary key, itemname varchar(25), rate
integer);")
cursor.execute("insert into item values('1005','Printer','8000');")
connection.commit()
connection.close()
print("Item table is created and a record is added Successfully")
```

### 5. பின்வரும் supplier மற்றும் item அட்டவணைகளை கவனித்து, (i) மற்றும் (ii) வினாக்களுக்கு பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

i) டெல்லியில் வசிக்காத மொத்த விற்பனையாளர்களின் Name, City மற்றும் Itemname களை திரையிடவும்.

ii) அகிலாவின் suppQty யில் உள்ள மதிப்போடு 40-யை அதிகரிக்கும்

| SUPPLIER |         |           |       |         |
|----------|---------|-----------|-------|---------|
| Suppno   | Name    | City      | Icode | SuppQty |
| S001     | Prasad  | Delhi     | 1008  | 100     |
| S002     | Anu     | Bangalore | 1010  | 200     |
| S003     | Shahid  | Bangalore | 1008  | 175     |
| S004     | Akila   | Hydrabad  | 1005  | 195     |
| S005     | Girish  | Hydrabad  | 1003  | 25      |
| S006     | Shylaja | Chennai   | 1008  | 180     |
| S007     | Lavanya | Mumbai    | 1005  | 325     |

```
import sqlite3
connection=sqlite3.connect("item.db")
cursor=connection.cursor()
```

i) டெல்லியில் வசிக்காத மொத்த விற்பனையாளர்களின் Name,City மற்றும் Itemname களை திரையிடவும்.

```
cursor.execute ("SELECT supplier.name, supplier.city, item.itemname FROM
supplier, item WHERE supplier.city <> 'Delhi' ")
ans=cursor.fetchall ()
for i in ans:
print(i)
```

ii) அகிலாவின் suppQty யில் உள்ள மதிப்போடு 40-யை அதிகரிக்கும்

```
cursor.execute ("UPDATE supplier SET supplier.SuppQty =SuppQty + 40
WHERE name='Akila' ")
connection.commit()
connection.close()
```

**பாடம்-16: தரவு காட்சிப்படுத்துதல் :PYPLOT பயன்படுத்தி கோட்டு வரைபடம், வட்ட உரைப்படம் மற்றும் பட்டை வரைபடம் உருவாக்குதல்**

**பகுதி - ஆ**

**அனைத்து வினாக்களுக்கும் விடையளி**

**(2 மதிப்பெண்)**

**1. தரவு காட்சிப்படுத்துதல் என்றால் என்ன?**

- தரவு காட்சிப்படுத்துதல் என்பது தரவு மற்றும் தகவல்களை வரைகலையாக உருவாக்குகின்றது.
- தரவு காட்சிப்படுத்தலின் முக்கிய நோக்கம் பயனாளர்களுக்கு தகவல்களை படக்காட்சி முறையில் காண்பிப்பது.

**2. பொதுவான தரவு காட்சிப்படுத்துதல் வகை யை பட்டியலிடுக.**

- விளக்கப்படங்கள்
- அட்டவணைகள்
- வரைகலை
- நிலப்படங்கள்
- இன்போகிராபிக்ஸ்
- டேஷ்போர்ட்

**3. Matplotlib ல் உள்ள காட்சிப்படுத்துதல் வகைகளை பட்டியலிடுக.**

- கோட்டு வரைவு
- ஸ்கேட்டர் வரைவு
- ஹிஸ்டோகிராம்
- பெட்டி வரைவு
- பட்டை விளக்கப்படம்
- வட்ட விளக்கப்படம்

**4. Matplotlib யை எவ்வாறு நிறுவலாம்?**

- Pip பயன்படுத்தி matplotlib நாம் நிறுவ முடியும்.
- Pip என்பது பைத்தான் தொகுப்புகளை நிறுவுவதற்கான ஒரு மேலாண்மை மென்பொருள் ஆகும்.
- python -m pip install -U matplotlib

**5. plt.plot([1,2,3,4]), plt.plot([1,2,3,4], [1,4,9,16]) ஆகிய இரு செயற்கூறுகளிடையேயான வேறுபாட்டை எழுதுக.**

**plt.plot([1,2,3,4])**

- matplotlib இதனை y மதிப்புகளின் தொடர் என கருதி தானாகவே, x அச்சின் மதிப்புகளை உருவாக்கும்.

- பைத்தான் வரம்புகள் 0 உடன் தொடங்கும் என்பதால், இயல்பாகவே x திசை y திசையை போன்று ஒரே நீளம் கொண்டதாக இருக்கும். எனவே x ன் தரவு மதிப்புகள் [0,1,2,3] என இருக்கும்.

`plt.plot([1,2,3,4], [1,4,9,16])`

- இந்த கட்டளை பல செயலுருபுகளை எடுத்துக்கொள்ளும். ஆனால் முதல் இரண்டு மதிப்பு 'x' மற்றும் 'y' ஆயத்தொலைவுகளை குறிக்கும்.
- இதன் பொருள், இந்த பட்டியல்களின் படி (1,1), (2,4), (3,9) மற்றும் (4,16) என்ற நான்கு ஆயத்தொலைவுகளைக் கொண்டிருக்கும்.

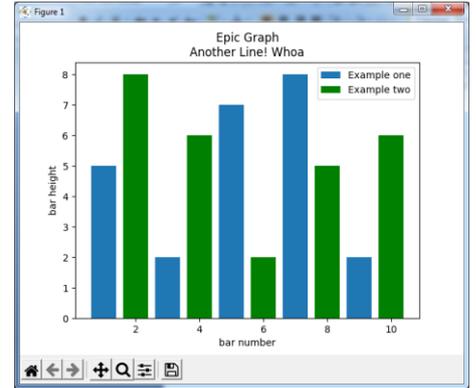
## பகுதி - இ

அனைத்து வினாக்களுக்கும் விடையளி

(3 மதிப்பெண்)

1. பின்வரும் தரவு காட்சிப்படுத்துதல் வரைவியின் வெளியீட்டை வரையவும்.

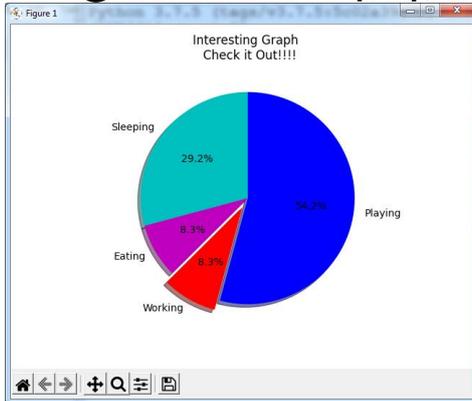
```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')
plt.legend()
plt.xlabel('bar number')
plt.ylabel('bar height')
plt.title('Epic Graph\nAnother Line! Whoa')
plt.show()
```



2. தரவு காட்சிப்படுத்தலின் மூன்று பயன்பாட்டை எழுதவும்.

- தரவு காட்சிப்படுத்தல் பயனர்கள் தரவுகளை எளிதாக கூர்ந்து ஆய்வு செய்யவும், உட்பொருளை வெளிப்படுத்த உதவுகிறது.
- இது சிக்கலான தரவுகளை புரிந்துக் கொண்டு அவற்றை பயன்படுத்திக் கொள்ளவழி செய்கிறது.
- தரவு காட்சிப்படுத்தல் பல்வேறு வரைப்படங்களைக் கொண்டு தரவு மாறிகளுக்கு இதையே உள்ள உறவு நிலையை வெளிப்படுத்துகிறது.

3. பின்வரும் வட்ட வரைப்படத்தை வெளியீடாக பெற குறிமுறையை எழுதவும்.



```
import matplotlib.pyplot as plt
sizes=[29.2,8.3,8.3,54.2]
labels=["Sleeping", "Eating", "Working", "Playing"]
cols=['c','m','r','b']
plt.pie(sizes, labels=labels, colors=cols,startangle=90,
 shadow=True,explode=(0,0,0.1,0), autopct="%1.1f%% ")
plt.title("Interesting Graph \n Check it Out!!!!")
plt.show()
```

## பகுதி- ஈ

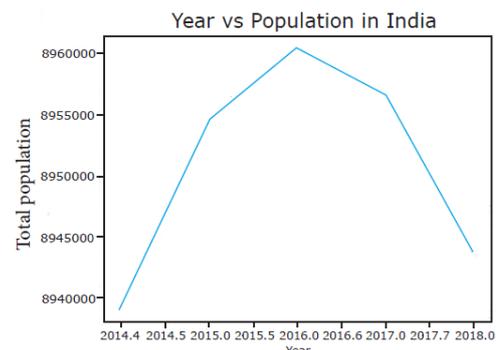
அனைத்து வினாக்களுக்கும் விடையளி

(5 மதிப்பெண்)

1. Matplotlib யை பயன்படுத்தும் pyplot வகைகளை விரிவாக விவரி.

கோட்டு வரைவு

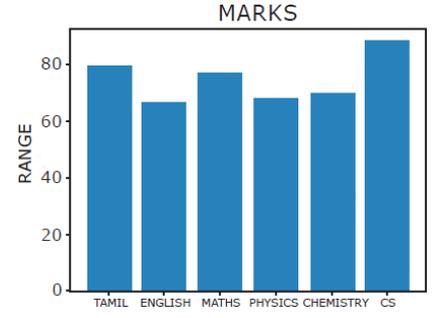
- கோட்டு வரைவு அல்லது கோட்டு விளக்கப்படம் என்பது தகவல்களை, "குறிப்பான்கள்" என்று அழைக்கப்படும் தரவு புள்ளிகளின் தொடரை நேர் கோட்டின் இணைப்பதன் மூலம் காட்டுகிறது.
- ஒரு கோட்டு வரைவு பொதுவாக கால இடைவெளி, நேரத் தொடர் போன்ற போக்கு பகுப்பாய்வுக்கான தரவுகளை காட்சிப்படுத்த பயன்படுகிறது.



- Matplotlib-ல் கோட்டு விளக்கப்படத்தை உருவாக்க, plt.plot() என்ற செயற்கூறினை பயன்படுத்த வேண்டும்.

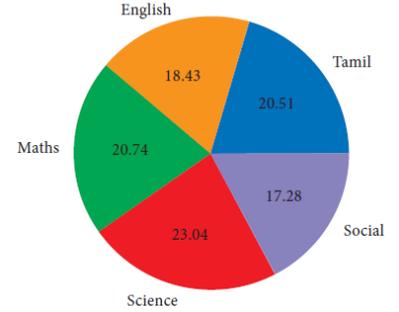
#### பட்டை விளக்கப்படம்

- பட்டை வரைவி (அல்லது) பட்டை விளக்கப்படம் என்பது மிகவும் பொதுவான வரைவிடம் வகையாகும்.
- இது எண் வகை தரவுகளுக்கும், மதிப்புகளுக்கும் இடையே உள்ள உறவை வெளிப்படுத்துகிறது.
- தரவுகளை செவ்வக பட்டையாக செங்குத்தாகவோ அல்லது கிடைமட்டமாகவோ காட்டும்.
- Matplotlib-ல் பட்டை விளக்கப்படத்தை உருவாக்க, plt.bar() என்ற செயற்கூறினை பயன்படுத்த வேண்டும்.



#### வட்ட விளக்கப்படம்

- வட்ட விளக்கப்படம் ஒரு பொதுவான விளக்கப்பட வகையாகும்.
- வட்ட வடிவில் தோன்றும் இந்த விளக்கப்படம் எண் விகிதத்தை விளக்கும் விதத்தில் துண்டுகளாக பிரிக்கப்பட்டிருக்கும்.
- வட்ட விளக்கப்படத்தில் உள்ள துண்டுகள் முழு படத்துடன் உள்ள உறவை வெளிக்காட்டும்.
- Matplotlib கொண்டு வட்ட விளக்கப்படத்தை உருவாக்க plt.pie() செயற்கூறினை பயன்படுத்த வேண்டும்.



### 2. Matplotlib திரையில் காணப்படும் பல்வேறு பொத்தான்களை விளக்கு.

|                                  |  |                                                                                                                                                                                                                                                      |
|----------------------------------|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| முகப்பு பொத்தான்                 |  | இப்பொத்தானை பயன்படுத்தி அசல் காட்சி திறையை எப்பொழுது வேண்டுமானாலும் பெறலாம்.                                                                                                                                                                         |
| முன்னோக்கி / பின்னோக்கி பொத்தான் |  | இப்பொத்தானைப் பயன்படுத்தி முந்தைய இடத்திற்கோ அல்லது முன்னோக்கி செல்லவோ முடியும்.                                                                                                                                                                     |
| பான் ஆக்ஸிஸ் பொத்தான்            |  | சிலுவை போன்ற தோற்றத்தை கொண்ட இப்பொத்தானை கிளிக் செய்து கொண்டே இழுத்து வரைபடத்தினுள் சுற்றி நகரலாம்.                                                                                                                                                  |
| பெரிதாக்கு பொத்தான்              |  | குறிப்பிட்ட ஒரு பகுதியை பெரிதாக்க (Zoom-in) இப்பொத்தானை கிளிக் செய்து சதுர வடிவில் இழுக்க அப்பகுதியை மட்டும் பெரிதாக்கப்படும். பெரிதாக்க இடது கிளிக் செய்து இழுக்க வேண்டும். சிறியதாக்க (Zoom-out) வலது கிளிக் செய்து (Right click) இழுக்க வேண்டும். |
| சப் - ப்ளாட் கட்டமைப்பு பொத்தான் |  | இப்பொத்தான், படம் மற்றும் வரைவிடத்திற்கு இடையே உள்ள இடைவெளியை கட்டமைக்க உதவுகிறது.                                                                                                                                                                   |
| படத்தை சேமிக்கும் பொத்தான்       |  | இப்பொத்தான் படங்களை பல்வேறு வடிவங்களில் சேமிக்க உதவும்.                                                                                                                                                                                              |

### 3. பின்வரும் செயற்கூறுகளின் பயன்பாட்டை எழுது:

(அ) plt.xlabel      (ஆ) plt.ylabel      (இ) plt.title      (ஈ) plt.legend()      (உ) plt.show()

(அ) plt.xlabel() → X-அச்சின் தலைப்பை குறிப்பிடுகிறது.

(ஆ) plt.ylabel() → Y-அச்சின் தலைப்பை குறிப்பிடுகிறது.

(இ) plt.title() → வரைப்படத்தின் தலைப்பை குறிப்பிடுகிறது.

(ஈ) plt.legend() → கொடாநிலை புனைவுகள் செயலாக்கலாம்.

(உ) plt.show() → வரைவியை காட்டலாம்.