NADAR HR.SEC.SCHOOL, RAJAPALAYAM.

XII - COMPUTER SCIENCE – TAMIL MEDIUM

QUARTERLY EXAM 2024 ANSWER KEY

PART - A

1.	பின்வரும் எது ஒரு பொருள் செய்ய வேண்டியதை தீர்மானிக்கிறது?	இடைமுகம்
2.	உருவமைப்பு அறியப்படாத தரவு வகை எவ்வாறு அழைக்கப்படுகிறது?	Abstract datatype
3.	எது வரையறுக்கப்பட்ட இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படும் உறுப்புகள் ஆகும்.	protected உறுப்புகள்
4.	செல்லுபடியாக்கும் உள்ளீட்டு எதிர்பார்க்கப்படும் வெளியீட்டைத் தரும் நெறிமுறை இவ்வாறு அழைக்கப்படுகிறது	நெறிமுறை சார் தீர்வு
5.	பின்வரும் எந்த குறியுரு பைத்தான் நிரலில் குறிப்புகளை உள்ளீடு செய்ய பயன்படுகிறது?	#
6.	பின்வருவனவற்றில் எது jump கூற்று கிடையாது?	for
7.	தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறை இவ்வாறு அழைப்பர்.	தற்சுழற்சி
8.	செயற்கூறுக்கு எந்த செயலுருபு சரியான இட வரிசையில் செயலுருபுகளை அனுப்பும்?	தேவையான
9.	பைத்தானில் சரங்களானது:	மாறாத்தன்மையுடையது
10.	stride என்பது பின்வருபவனவற்றுள் எதை குறிக்கும்?	slice செயற்பாட்டின் மூன்றாவது அளபுருவாகும்
11.	Let list 1 = [2, 4, 6, 8, 10), எனில் print(List1 [-2]) ன் விடை	8
12.	பின்வரும் எந்த பைத்தான் செயற்கூறு ஏற்கனவே உள்ள List -ல் ஒன்றுக்கும் மேற்பட்ட உறுப்புகளை சேர்க்கப் பயன்படுகிறது?	extend()
13.	பின்வரும் வழிமுறைகளில் எது அழிப்பியாகப் பயன்படுகிறது?	_ _del_ _()
14.	பொருளை உருவாக்கும் செயல்முறை எது:	சான்றுருவாக்கல்
15.	While மடக்கில் ------ பகுதி விருப்பத்தேர்வு	else

PART - B

16)	**துணை நிரல் என்றால் என்ன?** துணை நிரல்கள் கணினி மொழிகளின் அடிப்படைக் கட்டுமான தொகுதியாக விளங்குகின்றன. துணை நிரல்கள் என்பன ஒரு குறிப்பிட்ட செயலை மீண்டும் மீண்டும் செய்யப் பயன்படும் சிறிய நிரல் தொகுதியாகும். நிரலாக்க மொழிகளில் இத்துணை நிரல்கள் செயற்கூறுகள் (Functions) என்று அழைக்கப்படுகின்றன.
17)	**ஆக்கிகள் மற்றும் செலக்டர்கள் வேறுபாடு தருக.** ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது. செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.
18)	**private மற்றும் protected அணுகியல்புகளை பைத்தான் எவ்வாறு குறிப்பிடுகிறது.** பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக்கோடிடும் வழக்கத்தைப் பரிந்துரைக்கிறது. இதனால் private மற்றும் protected அணுகியல்பு வரையறுப்பிகள் சில பண்புகளைப் பின்பற்றுகின்றன.
19)	**1. முன்னறிப்பு (Priori Estimates) மதிப்பீடுகள் :** இது நெறிமுறையின் கோட்பாட்டு செயல்திறன் பகுப்பாய்வு ஆகும். வெளிப்புற காரணிகளை ஏற்றுக் கொண்டு நெறிமுறையின் செயல் திறன் அளக்கப்படுகிறது. **2. பின்னிய (Posteriori testing) சோதனை :** இது செயல்திறன் அளவிடுதல் ஆகும். இந்த பகுப்பாய்வில் உண்மையான புள்ளி விவரங்களான இயங்கும் நேரம் மற்றும் நிரல் நெறிமுறை நிறைவேற்ற தேவைப்படும் இடம் ஆகியவை சேகரிக்கப்படுகிறது.
20)	**பைத்தானில் உள்ள பல்வேறு செயற்குறிகள் யாவை?** பைத்தானில் உள்ள பல்வேறு செயற்குறிகள்: (i) கணித செயற்குறிகள் (ii) தொடர்புடைய அல்லது ஒப்பீடு செயற்குறிகள் (iii) தருக்க செயற்குறிகள் (iv) மதிப்பிருத்து செயற்குறிகள் (v) நிபந்தனை செயற்குறி

Kindly Send Me Your Key Answer to Our email id - Padasalai.net@gmail.com

21)	**break கூற்றைப் பற்றி குறிப்பு வரைக.** (i) break கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது. நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது. (ii) நிபந்தனையானது தவறு என்று பரிசோதிக்கும் வரை while அல்லது for மடக்கு செயல்படுத்தப்படும். ஆனால், break கூற்றைப் பயன்படுத்தி கட்டுப்பாட்டை மடக்கை விட்டு நிறுத்தி வெளியேறச் செய்யமுடியும். (iii) break கூற்று பின்னலான மடக்கின் உள்ளே இருந்தால், break கூற்று மிகவும் உள்ளே உள்ள மடக்கை விட்டு வெளியேறும்.
22)	**செயற்கூறுவின் முக்கிய நன்மைகள் யாவை?** செயற்கூறுவின் முதன்மை நன்மைகள் : (i) குறிமுறையை மீண்டும் எழுதுவதை தவிர்த்து குறிமுறையின் மறு பயனாக்கத்திற்கு உதவுகிறது. (ii) நமது பயன்பாட்டிற்குச் சிறந்த கூறுநிலையை வழங்குகிறது.
23)	**பைத்தானில் சரத்தை எவ்வாறு நீக்குவாய்?** சரத்தில் உள்ள ஒரு குறிப்பிட்ட குறியுருவை நீக்கவும் அனுமதிக்காது. ஆனால் del-கட்டளை பயன்படுத்தி ஒரு முழு சர மாறியையும் நீக்க முடியும்
24)	**ஒரு Tuples n எண்ணிக்கை உறுப்புகளுடன் உருவாக்குவதற்கான தொடரியலை எழுதுக.** **விடை.** Tuples_Name=(e1,e2,e3…….en) Tuples_Name=e1,e2,e3…….en Tuples_Name=(e1,)
	PART - C
25)	**pure செயற்கூறு** 1. pure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்தே அமையும். 2. அதனால், pure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் எப்பொழுதும் அதே திருப்பி அனுப்பும் மதிப்பே கிடைக்கும் இது எந்த பக்க விளைவுகளையும் கொண்டிருக்காது. 3. இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யும். **impure செயற்கூறு** 1. impure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்து அமையாது, 2. அதனால் Impure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் வெவ்வேறான திருப்பி அனுப்பும் மதிப்பு கிடைக்கும். 3. இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யாது.
26)	**முழுதளாவிய வரையெல்லையை எடுத்துக்காட்டுடன் விவரி.** முழுதளாவிய வரையெல்லை நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும். அதாவது, முழுதளாவிய மாறிகளை நிரலின் அனைத்து செயற்கூறுகளுக்கு உட்புறமும், வெளிப்புறமும் அணுக முடியும். பின்வரும் எடுத்துக்காட்டைக் காண்போம்.
27)	**நெறிமுறையின் பண்பியல்புகளைப் பட்டியலிடுக.** (i) உள்ளீடு (ii) வெளியீடு (iii) எல்லையற்றது (iv) வரையறுத்தல் (v) செயல்தன்மை (vi) உண்மைத் தன்மை (vii) எளிமை (viii) குழப்பமற்றது (ix) செயலாக்கம் (x) அடக்கமானது (xi) சார்பற்றது
28)	**மும்ம செயற்குறியை எடுத்துக்காட்டுடன் எழுதுக.** மும்ம செயற்குறி நிபந்தனை செயற்குறி என்றும் அழைக்கப்படுகிறது. இது சமன்பாடுகளின் நிபந்தனையை சரி அல்லது தவறா என்று சோதித்து செயல்படுத்தும், மும்ம செயற்குறி பல வரி if..else கூற்று போல் அல்லாது நிபந்தனைகளை ஒற்றை வரியில் சோதிக்க அனுமதிக்கிறது. நிபந்தனைச் செயற்குறியின் தொடரியல், Variable Name = [on_true] if [Test expression) else [on_false] **எடுத்துக்காட்டு:** min=50 if 49 > 50 else 70

29)	**Program :**
	for i in range(65, 70):
	    for j in range(65, i+1):
	        print(chr(j), end='\t')
	    print('\n')

30) **உள்ளமை மாறிகளுக்கான விதிமுறைகளை எழுதுக.**

உள்ளமை மாறியின் விதிமுறைகள் :

(i) உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.

(ii) செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.

(iii) செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உருவாக்கப்படும்.

31) **பைத்தானில் count( ) செயற்கூறு பற்றி குறிப்பு வரைக.**

count(str, beg, end)	ஒரு சரத்தில் கொடுக்கப்பட்ட பரப்பிற்குள் உள்ள துணை சரங்களின் (substring) எண்ணிக்கையை திருப்பும். ஒரு துணை சரம் என்பது ஒற்றை குறியுருவாக கூட இருக்கலாம். பரப்பு அளபுருக்களை கொடுக்க வேண்டியது கட்டாயமில்லை கொடுக்காவிட்டால் பைத்தான் முழு சரத்தையும் தேடும். தேடல் எழுத்துணர்வு மிக்கது (case sensitive) என்பதை நினைவில் கொள்க	>>> str1="Raja Raja Chozhan"
		>>> print(str1.count('Raja'))
		2
		>>> print(str1.count('r'))
		0
		>>> print(str1.count('R'))
		2
		>>> print(str1.count('a'))
		5
		>>> print(str1.count('a',0,5))
		2
		>>> print(str1.count('a',11))
		1

32)

```
def __init__(self, [args ……..]):
<statements>

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

33) **பின்வரும் குறிமுறையின் வெளியீடு என்ன?**

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

வெளிப்பாடு : [1,2,4,8,16]

PART - D

34) A) **1. செயலுருபுகள் என்றால் என்ன?**
(அ) தரவுவகை இல்லாத அளபுருக்கள்
(ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் விவரி.

விடை: செயலுருபுக்கள்: செயலுருபுக்கள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.

அ) தரவு வகை இல்லாத அளபுருக்கள் : செயற்கூறு வரையறைக்கு ஓர் எடுத்துக்காட்டைக் காணலாம்.

```
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. நாம் எந்த தரவினத்தையும் குறிப்பிடவில்லை என்பதை நினைவில் கொள்க.

* சில மொழிகளின் நிரல் பெயர்ப்பி இவ்வகை சிக்கல்களை நிரல் நெறிமுறைப்படி சரி செய்கிறது. ஆனால் சில நிரல் பெயர்ப்பிக்கு தரவு வகையைக் குறிப்பிடுவது கட்டாயமாகும்.

* 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. if கோவையின் தரவு வகை int ஆக இருப்பதால், செயற்கூறின் திருப்பி அனுப்பும் மதிப்பும் int ஆக இருக்கும், 'b' யின் மதிப்பு சுழியத்தோடு = செயற்குறியுடன் ஒப்பீடு செய்யப்படுகிறது.

* அதனால் 'b' யின் தரவுவகையும் 'int' ஆகும். * செயற்குறியுடன் 'a' யின் மதிப்பு மற்றொரு கோவையோடு பெருக்குத் தொகையைக் கணக்கிடுவதால், 'a' யின் வகையும் int ஆகும்.

ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் : சில காரணங்களுக்காக, இப்பொழுது நாம் அதே செயற்கூறு வரையறை தரவு வகையுடன் எழுதலாம்.

```
let rec pow(a: int) (b: int) : int: =
if b=0 then I
else a * pow b(a-1)
```

* 'a' மற்றும் 'b' தரவு வகை குறிப்பு (type annotations) எழுதும் போது, அடைப்புக்குறிக்குள் ( ) அவசியமானது ஆகும். பொதுவாக, இந்த குறிப்புகளை நாம் விட்டுவிடலாம்.

* ஏனெனில், நிரல்பெயர்ப்பி இவற்றை அனுமானிப்பது மிகவும் எளிது. முன்பெல்லாம் நாம் வெளிப்படையாகவே தரவு வகைகளை எழுதுவோம்.

* எந்த வித அர்த்தமும் இல்லாத தரவு வகை பிழைச் செய்தியைப் பெறும் போது, இது மிகவும் பயனுள்ளதாகும். தரவு வகைக்கு வெளிப்படையாக தரவுவகை குறிப்பு எழுதுவது பிழைச் செய்தியைத் திருத்தம் செய்வதற்கு பயனுள்ளதாக இருக்கும்.

34) B)	**தரவு அருவமாக்கம் எவ்வாறு செயல்படுத்துவாய்? எடுத்துக்காட்டுடன் விளக்குக.** தரவு அருவமாக்கினை செயல்படுத்த, ஆக்கிகள் (Constructor) மற்றும் செலக்டர் (Selectors) என்ற இரண்டு செயற்கூறுகள் உருவாக்கப்பட வேண்டும். **ஆக்கிகள் மற்றும் செலக்டர்கள் (constructors and selectors) :** (i) ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது. செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது. (ii) 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) (iii) city பொருளின் தகவல்களை பெறுவதற்கு பின்வரும் செயற்கூறுகளை பயன்படுத்தலாம். * getname(city) * getlat(city). * getlon(city) (iv) பின்வரும் போலி குறிமுறை இரு நகரங்களுக்கு இடையேயான தொலைதூரத்தை கணக்கிடும். distance(city1, city2): lt1, lg1 := getlat(city1), getlon(city1) lt2, lg2:= getlat (city2), getlon (city2) return (1t1 - 1t2)**2 + (1g1-1g2)**2)1/2 (v) மேலே காணும் குறிமுறையில், distance(),getlat() மற்றும் getlon() ஆகியவை செயற்கூறுகள் ஆகும். lt என்பது அட்ச ரேகை மற்றும் 1g என்பது தீர்க்க ரேகையும் குறிக்கிறது. longitude. := என்பதை "assigned as" அல்லது "becomes" என்று வாசிக்க வேண்டும். lt 1, lg1 := getlat(city1), getlon(city1) (vi) இதனை 1t1 என்பது getlat(city1)ன் மதிப்பாகிறது மற்றும் 1g1 என்பது getlont(city1) ன் மதிப்பாகிறது. என்று வாசிக்க வேண்டும். (vii) இந்த செயற்கூறுகள் எவ்வாறு செயல்படுத்தப்படுகிறது என்பதை தெரிந்து கொள்ள வேண்டியதில்லை. இதை வேறு ஒருவர் நமக்காக வரையறுத்துள்ளார் என்று கருதிக் கொள்ள வேண்டும். பயனர் செயற்கூறுகள் எப்படி செயல்படுத்தப்படுகின்றது என்பது தெரிந்து வைத்திருக்க வேண்டியதில்லை எனினும் வேறு ஒருவரால் இச்செயற்கூறுகள் வரையறுக்கப்பட்டிருக்க வேண்டும். (viii) மேலே காணும் குறிமுறையில், ஆக்கி மற்றும் செலக்டர்களை அடையாளம் காண்போம். கண்ஸ்டரக்டர்ஸ் செயற்கூறுகள் அருவமாக்க தரவு வகையை கட்டமைக்கப் பயன்படுகிறது என்பதை நாம் அறிவோம். மேலே காணும் போலிக் குறிமுறையில், cityயின் பொருளை உருவாக்கும் செயற்கூறு, ஆக்கி ஆகும். city = makecity(name, lat, lon) இங்கு makecity(name, lat, lon) என்ற ஆக்கி city எனும் பொருளை உருவாக்குகிறது. (ix) செலக்டார் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது. மேலே காணும் குறிமுனையில், * getname(city) * getlat(city) * getlon(city) என்பவை city எனும் பொருளிலிருந்து தகவல்களை பெற்றுத் தரும் செலக்டர் செயற்கூறுகளாகும்.

www.Padasalai.Net

35) A)

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.

அடைக்கப்பட்ட வரையெல்லை:

- 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.

முழுதளாவிய வரையெல்லை:

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

உள்ளிணைந்த வரையெல்லை:

- நிரல்பெயர்ப்பி அல்லது தொகுப்பானை தொடங்கும் பொழுது உள்ளிணைந்த வரையெல்லையனானது நிரல் வரையெல்லையில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும்.
- நிரலாக்க மொழியின் நூலக செயற்கூறினுள் வரையறுக்கப்பட்ட மாறி அல்லது தொகுதி உள்ளிணைந்த வரையெல்லையைக் கொண்டுள்ளது.

எடுத்துக்காட்டு:

```
Built-in/module scope    →    நூலக கோப்புகள்
a := 10                  →    முழுதளாவிய வரையெல்லை
Disp():
  b := 7                 →    அடைக்கப்பட்ட வரையெல்லை
  Disp1()
     c := 5              →    உள்ளமை வரையெல்லை
     print a, b, c
  Disp1()
Disp()
```

நிரலின் வெளிப்பாடு

```
10
7
5
```

35) B)

இருமத் தேடல் என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.

இருமத் தேடல் :

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

இருமத் தேடலுக்கான போலி குறிமுறை :

1. மைய உறுப்பிலிருந்து தொடங்கவும் :

(i) இலக்கு மதிப்பும் அணியின் மைய உறுப்பும் நிகர் எனில் (அதாவது, மைய இலக்கு = உறுப்புகளின் எண்ணிக்கை /2) மைய உறுப்பின் சுட்டெண்ணைத் திருப்பி அனுப்பும்.

(ii) நிகரில்லை என்றால், மைய உறுப்பை மதிப்போடு ஒப்பிடவும்.

(iii) மைய சுட்டெண்ணிலுள்ள எண் இலக்கு மதிப்பை விட பெரியது எனில், மைய சுட்டெண்ணிற்கு வலப்புறம் உள்ள உறுப்புகளைத் தேர்ந்தெடுத்து படிநிலை -1லிருந்து தொடங்கவும்.

(iv) 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 :

எடுத்துக்காட்டு

www.Padasalai.Net

36) A)

input() மற்றும் output() செயற்கூறுகள் பற்றி விளக்கு.

உள்ளீடு மற்றும் வெளியீடு செயற்கூறுகள்

- ஒரு நிரல் பயனர் விரும்பிய பணியை நிறைவேற்றுவதற்கு பயனருடன் தொடர்பு கொள்ள வேண்டும். இதற்கு உள்ளீடு - வெளியீடு செயற்கூறுகள் பயன்படுகிறது.
- input() செயற்கூறு, ஓர் நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடு செய்யவும்,
- print() என்ற வெளியீடு செயற்கூறு நிரலின் தீர்வுகளை திரையில் காண்பிக்க உதவுகிறது.

Print() செயற்கூறு

- பைத்தானில், print() செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை வெளியிட பயன்படுகிறது.
- Print() செயற்கூறு தீர்வுகளை திரையில் காண்பிக்கும் முன் கோவையை மதிப்பீடு செய்யும்.
- print() செயற்கூறினுள் கொடுக்கப்படும் முழு கூற்றினையும் திரையில் காட்டும்.
- காற்புள்ளி(,) கொண்டு print() செயற்கூறினுள் ஒன்றுக்கு மேற்பட்ட உருப்புகளை பிரிக்கலாம்.
- print() செயற்கூறின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது

```
print ("string to be displayed as output")
print (variable)
print ("String to be displayed as output", variable)
print ("Stringl", variable, "String 2", variable, "String 3" ........)
```

எடுத்துக்காட்டு:

```
>>>print("Welcome to Python Programming")
 Welcome to Python Programming
```

input() செயற்கூறு

பைத்தானில், input() செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடாக பெற்றுக் கொள்ளப் பயன்படுகிறது. input() செயற்கூறின் தொடரியல்,

```
Variable = input ("prompt string")
```

- இந்த தொடரியலில் உள்ள "prompt string" பயனர் கொடுக்கப்பட வேண்டிய உள்ளீடு எதுவென்பதை உணர்த்தும் கூற்று அல்லது செய்தியை கொண்டிருக்கும்.
- input() கட்டளை அமைப்பில் உள்ள "prompt string"திரையில் தோன்றும், பயனர் இதை படித்து உள்ளீடு சாதனத்திலிருந்து எதிர்பார்க்கப்படும் தரவுகளை கொடுக்கலாம்.
- input() செயற்கூறு, விசைப்பலகையின் மூலம் தட்டச்சு செய்தவற்றை மாறியில் சேமித்து வைக்கும்.
- "prompt string" கொடுக்கப்படவில்லையெனில் திரையில் எந்த தகவலும் தோன்றாது. இதனால் பயனர் எவற்றை உள்ளீடு செய்ய வேண்டும் என்பதை தெரிந்து கொள்ள முடியாது.
- Print() செயற்கூறு தீர்வுகளை திரையில் காண்பிக்கும் முன் கோவையை மதிப்பீடு செய்யும்.

```
>>> city=input ("Enter Your City:")
 Enter Your City: Madurai
```

36) B)

if..else..elif கூற்றை எடுத்துக்காட்டுடன் விளக்குக.

பின்னலான if..else..elif கூற்று :

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
```

'elif' பகுதி if..else-if..else கூற்றுகளை இணைக்கு ஒன்றிணைத்து ஒரே if..elif...else...... 'elif' ன் விரிவாக்கம் else if. if கூற்றில் பயன்படுத்தப்படும் 'elif பகுதிக்கு வரம்பு இல்லை , ஆனால் else பகுதி பயன்படுத்தும்போது கூற்றின் இறுதியில் பயன்படுத்த வேண்டும்.

எடுத்துக்காட்டு

```
If a>b and a>c:
        Print("a is big")
elif b>c:
        Print("b is big")
else:
        Print("c is big")
```

www.Padasalai.Net

37) A)

பின்வரும் உள்ளிணைந்த செயற்கூறுகளை விளக்குக.

(a) id() (b) chr() (c) round() (d) type () (e) pow ()

செயற்கூறு	விளக்கம்	பொதுவடிவம்	எடுத்துக்காட்டு
id ( )	கொடுக்கப்பட்ட பொருளின் நினைவக முகவரியைத் திருப்பி அனுப்பும். குறிப்பு: x, y நினைவக முகவரி ஒவ்வொரு கணிப்பொறிக்கும் வேறுபடும்.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) வெளியீடு address of x is : 1357486752 address of y is : 13480736
chr ( )	கொடுக்கப்பட்ட ASCII மதிப்பிற்கு யுனிக்கோடு எழுத்தை திருப்பி அனுப்பும். இது ord() செயற்கூறின் தலைகீழாகும்.	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) வெளியீடு A +
round ( )	கொடுக்கபட்ட எண்ணிற்கு அருகே உள்ள முழு எண்ணாக மாற்றி திருப்பி அனுப்பும். 1. முதல் செயலுருபு கொடுக்கப்பட்ட எண்ணைக் குறிக்கும் 2. இரண்டாவது செயலுருபு முழு எண்ணிற்கு பிறகு தசம புள்ளிகளின் எண்ணிக்கையைக் குறிக்கும்.	round (number [,ndigits])	x= 17.9 y= 22.2 z= -18.3 print ('x value is rounded to', round (x)) print ('y value is rounded to', round (y)) print ('z value is rounded to', round (z)) வெளியீடு:1 x value is rounded to 18 y value is rounded to 22 z value is rounded to -18 n1=17.89 print (round (n1,0)) print (round (n1,1)) print (round (n1,2)) வெளியீடு:2 18.0 17.9 17.89
type ( )	கொடுக்கப்பட்ட பொருளின் தரவின வகையைத் திருப்பி அனுப்பும். குறிப்பு: ஒரு அளபுரு பொருளுடன் இது பயன்படுத்தப்படுகிறது.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) வெளியீடு <class 'float'> <class 'str'> <class 'bool'>
pow ( )	கொடுக்கப்பட்ட எண்ணின் ab அடுக்கு பெருக்கத்தை திருப்பி அனுப்பும். (a**b) ன் அடுக்கு b.	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) வெளியீடு: 25 125.0 343

37) B)

பைத்தானில் பயன்படும் சர செயற்குறிகளை தகுந்த எடுத்துக்காட்டுடன் விளக்குக.

பைத்தான் பின்வரும் செயற்குறிகளை வழங்குகிறது. இச்செயற்குறிகள் சரங்களை கையாள உதவுகிறது.

இணைப்பு (Concatenation +)

இரண்டு அல்லது அதற்கு மேற்பட்ட சரங்களை இணைக்கும் செயல்பாடு சேர்த்தல்/இணைத்தல் எனப்படும். கூட்டல் செயற்குறியானது சரங்களை பைத்தானில் இணைத்துக் கொள்ளப் பயன்படுகிறது.

எடுத்துக்காட்டு :

```
>>> "welcome" + "Python"
'welcomePython'
```

சேர்த்தல் (Append +=) :

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

பலமுறை (Repeating (*)) :

பெருக்கல் செயற்குறி கொடுக்கப்பட்ட சரத்தினை பல தடவைகள் வெளிப்படுத்த பயன்படுகிறது.

எடுத்துக்காட்டு:

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

38) A) பைத்தானிலுள்ள பல்வேறு set செயல்பாடுகளை பொருத்தமான எடுத்துக்காட்டுகளுடன் விளக்குக.

கணிதத்தில் கற்ற Set செயற்பாடுகள் ஆகிய ஒட்டு (Union), வெட்டு (intersection) வேறுபாடு (difference) சமச்சீரான வேறுபாடு (Symmetric difference), போன்றவற்றை பைத்தானிலும் பயன்படுத்தலாம்.

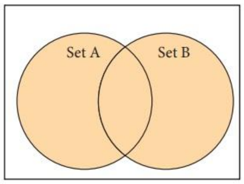
ஒட்டு (Union):

இது இரண்டு அல்லது அதற்கு மேற்பட்ட Set-களின் அனைத்து உறுப்புகளையும் உள்ளடக்கும்.

(i) பைத்தானில் | என்ற செயற்குறி இரண்டு Set களின் ஒட்டை உருவாக்கப்பயன்படுகிறது. Union செயற்கூறும் பைத்தானில் இரண்டு Set. களை இணைக்கப் பயன்படுகிறது.

(ii) எடுத்துக்காட்டு : ஒட்டு (Union) செயற்குறியைப் பயன்படுத்தி இரண்டு Set களை இணைப்பதற்கான நிரல்

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



வெளியீடு

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

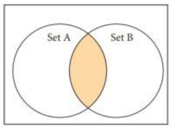
வெட்டு (intersection) :

இது இரண்டு Setகளின் பொதுவான உறுப்புகளை உள்ளடக்கியது.

(i) பைத்தானில் & செயற்குறி இரண்டு Set களை வெட்டுவதற்கு பயன்படுகிறது. வெட்டு (intersection()) செயற்கூறும் பைத்தானில் இரண்டு Set களை வெட்டுவதற்கு பயன்படுகிறது.

எடுத்துக்காட்டு : வெட்டு intersection() செயற்குறியைப் பயன்படுத்தி இரண்டு Setகளை வெட்டுவதற்கான நிரல்

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



வெளியீடு : {'A', 'D'}

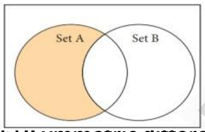
வேறுபாடு Difference:

இது முதல்Set(A)ல் உள்ள அனைத்து உறுப்புகளையும் உள்ளடக்கியது. இது இரண்டாவது Set-ஐ தவிர்க்கிறது.

(i) பைத்தானில் -(minus) செயற்குறி Set செயற்பாட்டின் வேறுபாட்டைக் கண்டறிய பயன்படுகிறது. difference() செயற்கூறும் வேறுபாட்டு செயற்பாட்டிற்காக பயன்படுகிறது.

(ii) எடுத்துக்காட்டு : செயற்குறியைப் பயன்படுத்தி இரண்டு setகளின் வேற்றுமைக்கான நிரல்

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



வெளியீடு : {2, 4}

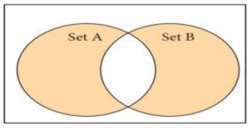
சமச்சீரான வேறுபாடு (Symmetric difference):

இது இரண்டு Set-ல் உள்ள பொதுவான உறுப்புகளை மட்டும் தவிர்த்து மற்ற அனைத்து உறுப்புகளையும் உள்ளடக்கியது.

(i) Caret(^) செயற்குறி பைத்தானில் சமச்சீரான வேறுபாட்டை கண்டறிய பயன்படுகிறது. Symmetric difference() செயற்கூறும் அதே செயலை செய்ய பயன்படுகிறது.

(ii) எடுத்துக்காட்டு : Caret (^) செயற்குறியைப் பயன்படுத்தி சமச்சீரான வேறுபாட்டை கண்டறியும் நிரல்.

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



வெளியீடு : {2, 4, 'B', 'C'}

38) B)

```
a = float(input('Enter first side: '))
b = float(input('Enter second side: '))
c = float(input('Enter third side: '))
s = (a + b + c) / 2
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```