

தமிழில் நிரல் எழுது எழில் நிரலாக்க மொழி

Write Code in Tamil

Ezhil Programming Language

சிறப்புச் சொற்கள் உதவி (Keywords Helper)

பதிபி	தோர்ந்தெடு	தோவு	ஏதேனில்	ஆனால்	இல்லைஆன
பின்கொடு	முடி	நிரல்பாக்கம்	தொடர்	நிறுத்தது	ஒவ்வொன்ற
>=	<=	!=	!	*	/
,	()	[]	%	^	

array2.n ▾ உதாரணம் பார்க்க (Browse Examples)

```
1 # (C) முத்தையா அண்ணாமலை 2013
2 # இந்த ஒரு எழில் தமிழ் நிரலாக்க மொழி உதாரணம்
3
4 z = [ 1 , 2 , 4, 5, 10 ]
5 x = [ 5-6, 8-10, 8*4 - 4*9, -5, 9 -11 - 8 ]
6
7 i = 0
8 @( i < len(x) ) வரை
9     பதிபி எடு(z,i) + எடு(x,i)
10     assert( ( எடு(z,i) + எடு(x,i) ) == 0 )
11     i = i + 1
12 முடி
13
14 பதிபி "வணக்கம்!", x
15
```

முத்தையா அண்ணாமலை,
என். சொக்கன்



தமிழில் நிரல் எழுது

- எழில் நிரலாக்க மொழி

Write Code in Tamil

- Ezhil Programming Language

முத்தையா அண்ணாமலை,

மற்றும்

என். சொக்கன்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

Copyright © 2013, 2015, முத்தையா அண்ணாமலை

பதிப்பாளர்: முத்தையா அண்ணாமலை

Typeset in Libre Office 3. Written in Tamil, using the UTF-8 Unicode encoding

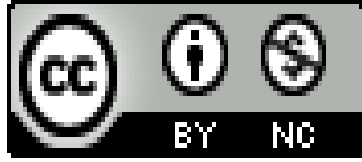
Published in Boston, MA, USA .



All Rights Reserved

This book may be used under the terms of the following license.

CC Attribution Non-Commercial 4.0 , International



This license lets others remix, tweak, and build upon your work non-commercially, and although their new works must also acknowledge you and be non-commercial, they don't have to license their derivative works on the same terms.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

License Deed: <http://creativecommons.org/licenses/by-nc/4.0/>

Legal Code: <http://creativecommons.org/licenses/by-nc/4.0/legalcode>

எழுத்தாளர்கள்



முத்தையா அண்ணாமலை

எழில் மொழியை உருவாக்கியவர்.
பாஸ்டன் நகரில் வசிக்கும் விஞ்ஞானி,
மற்றும் மென்பொருள் பொறியாளர்.



என். சொக்கன்

தமிழ் எழுத்தாளர். பொங்கனூர் நகரில்
வசிக்கும், IT-துரையில், தொழில்நுட்ப
ஆலோசகர். இவர் பல தமிழ்
புத்தகங்களை எழுதியுள்ளார்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

ஒப்புக்கைகளை

எழில் மொழியை துனிந்து தன்னார்வவலர்களாக முதலில் பயனபடுத்தியவர்களுக்கும், புத்தகத்தை பிழைதிருத்தியவர்களுக்கும் ஆழ்ந்த நன்றி. எழில் மொழி தமிழ் மாணவர்களை அடைய உதவிய நண்பர்களுக்கு நன்றி. அவர்கள் - அருண்ராம், டி.புனீனிவாசன், சாலா, சத்தியா, ஸ்ரீதர்; நண்பர்கள் மற்றும் உறவினர்கள்.

அர்ப்பணம்

பேராசிரியர் திரு. மெய்யப்பன், சாரதா ஆச்சி நினைவிற்கு
-அ. மு.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

முன்னுரை

தாய் மொழியில் கற்கும் குழந்தைகள் நன்கு புரிந்துணர்ந்து ஆழமாக கற்கிறார்கள் எனப் பல்வேறு ஆராய்ச்சிகள் கூறுகின்றன. இதுநாள் வரை, நம் தாய் மொழியான தமிழில் நிரலாக்கம் கற்பது என்பது ஓர் எட்டா கனவாகவே இருந்து வந்தது. அக்கனவை நனவாக்குவதுதான் "எழில்".

நமக்கு, அன்றாட வாழ்வில் பல்வேறு கணக்குகள், செயல்கள் செய்ய வேண்டி உள்ளன. அவற்றில் பலவற்றை நம்மால் கணினியின் உதவி கொண்டு எளிதாக செய்ய முடியும். ஆனால், இன்றைய தேதியில் கணினியோடு உரையாட, கணினியால் புரிந்து கொள்ள தெரிந்த மொழியை நாம் தெரிந்து வைத்திருக்க வேண்டும். அம்மொழிகள்தாம், C, C++, Java, Python போன்ற நிரலாக்க மொழிகள் (Programming Languages). இவை அனைத்தும் ஆங்கிலத்தை அடிப்படையாகக் கொண்டவை!

கணினியோடு உரையாட நாம் மொழிகளைக் கற்க வேண்டுமா? கொஞ்சம், வித்யாசமாக யோசிப்போம்! நாம் தமிழ் பேசுகிறோம். கணினிக்கு நாம் பேசும் தமிழைக் கற்று கொடுத்து விட்டால், எளிதில் நாம் தமிழிலே உரையாடி விடலாம் அல்லவா! இப்புதிய யோசனையின் பலன்தான் "எழில்"!

கணினி "எழில்" மூலம் தமிழ் பேசி நிரலாக்கம் செய்ய வரவேற்கிறது! நம் மொழியை கணினி கற்றுக் கொண்டுள்ளது! இனி, இதன் மூலம் நிரலாக்கம் செய்து, கணினி பேசும் மற்ற மொழிகளையும் நாம் கற்போம்!

ஒரு மேம்பட்ட மேல் நிலை (advanced high level) நிரலாக்க மொழியில், என்னென்ன அம்சங்கள் உள்ளனவோ, அவற்றில் பல (நிபந்தனைகள், தொடர் கணக்கீட்டு வாக்கியங்கள், நிரல் பாகங்கள், வரைதல், கோப்பு மாற்றி வாக்கியங்கள் போன்ற) முக்கிய அம்சங்கள், "எழில்" மொழியில் உள்ளன.

ஒரு நிரலாக்க மொழிக்கும் பயன்பாட்டாளருக்கும் உள்ள உறவு பரஸ்பர புரிதலையும் வளர்ச்சியையும் சார்ந்தது. ஒரு நிரலாக்க மொழியின் வளர்ச்சி, அம்மொழி எவ்வளவு எளிதாக எத்தனை பயன்பாட்டாளர்களின் தேவைகளை பூர்த்தி செய்கிறது என்பதிலும் பயன்பாட்டாளர்கள் உருவாக்கும் தொகுப்புகளிலும்தான் (libraries and packages) உள்ளது.

அவ்வகையில், நாம் "எழில்" மொழியில் தொகுப்புகள் பலவும் உருவாக்கி, கணினிக்கு

நாம் தமிழில் இன்னும் பல புதிய கட்டளைகள் இடுவோம்! இடும் கட்டளைகளை புரிந்து
கொள்ள கணினிக்கு கற்று கொடுப்போம்!

"எழில்" என்ற புதிய உலகத்திற்கு உங்களை அழைக்கிறோம்!

திரு. அருண்ராம்,
மின்னியல் பொறியாளர்
டவிட்டர்: @arunatma
பொங்களுர், 2015

பொருளடக்கம் (Table of Contents)

1. 1 -1 -வாங்க, நிரல் எழுதுவோம்!.....	1
1.1 முதல் நிரலை எப்படி எழுதுவது.....	1
1.1 களத்தை, கணினி மொழியைத் தீர்மானிப்பது.....	2
1.1.1. ஆனால், அதை எங்கே எழுதுவது?.....	2
1.1.2. என்ன நிரல் எழுதுவது எனத் தீர்மானிப்பது.....	2
1.1.3. நிரல் வழிமுறை (Algorithm).....	2
1.1.4.மாதிரி நிரல் எழுதுவது.....	3
1.1.5. முழுமையான நிரல் எழுதுவது.....	4
1.1.6.நிரலைப் பரிசோதிப்பது.....	5
1.1.7.நிரலைச் சேமித்துவைப்பது.....	5
2. 2 -2 -கணினி மென்பொருள் நிரல் எழுதக் கற்றுக்கொள்.....	7
2.1"எழில்" மொழியைக் கொண்டு நாம் என்னவெல்லாம் செய்யலாம்?.....	8
2.2 எழில் அறிமுகம் மற்றும் வரலாறு.....	8
2.3 எழில் மொழியின் நோக்கங்கள்:.....	9
2.4 குறிச்சொற்கள்:.....	9
3. 3 -3 -எழில் பதிவிறக்கம் மற்றும் நிறுவ (Installing Ezhil).....	11
3.1 வணக்கம் தமிழகம்!.....	11
3.2 இணையத்தை பயன்படுத்தும் எழில்.....	12
3.3 எழில் - பைத்தான் வழியில்.....	12
3.4 சில எளிய எழில் கணக்குகள்.....	13
4. 4 -4 - சரங்கள் மற்றும் எழுத்துகள் (Strings).....	16
4.1 சரம் நீளம் (String length).....	16
4.2 சரம் இணைவதற்கு (String length).....	16
4.3 தேடல், சரங்களை மாற்றவும் (Search Strings).....	18
4.4 பட்டியல்கள் பயன்படுத்த, வரிசைப்படுத்த (Using Lists, Sorting).....	18
5. 5 -5 -நிபந்தனை (Conditionals).....	21
5.1 ஆம் இல்லை எனில்.....	21
5.2 பயிற்சி.....	23
6. 6 -6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements).....	25
6.1 வரை குறிச்சொல் (While statement).....	25
6.2 தேர்ந்தெடு, நிறுத்து, தொடர் - குறிச்சொல் (Switch-Case, Break, Continue, statement).....	27
6.3 பயிற்சி.....	29
6.4 ஒவ்வொன்றாக குறிச்சொல் (For-Each statement).....	30
7. 7 -7 -கொஞ்சம் 'வரைந்து' பார்ப்போமா? (Drawing pictures - Turtle graphics).....	32
7.1 வலது, இடது வரைதல் - (Drawing Forward, Backward, Right, Left).....	33
7.2 எழுதுகோல்மேலே (Pen up, Pen down.....	33
7.3 நிறங்கள், வட்டங்கள் Colors, Circles	34
8. 8 -8 -நிரல்பாகம் (Functions – Composition).....	37
8.1 தொடர்பெருக்கு (Factorial).....	38

8.1.1.இந்த நிரல் எப்படி இயங்குகிறது?.....	39
8.2 மீண்டும் அழைக்கப்படும் நிரல்பாகம் (Recursion).....	39
8.3 பயிற்சி - பகா எண்கள் (Prime Numbers).....	41
9. 9 -9 -கோப்பு (File I/O).....	45
9.1 புதிய கோப்பு.....	45
9.1.1.கோப்பை மூட.....	45
9.2 கோப்பைத் திற படிக்க.....	46
9.3 முடிவு.....	47
10.....10 -10 -எழில் மொழி உதாரணங்கள் (Examples)	48
10.1 உதாரணங்கள் பதிவிறக்கம்.....	48
10.2 அறிமுகப்படுத்துதல்.....	48
10.3 அச்சிடுதல்	48
10.3.1.எழுத்துச் சரங்களை அச்சிடுதல்:	49
10.4 நிரலில் உள்ள விவரங்களை அச்சிடுதல்	49
10.5 உள்ளிடுதல்	50
10.6 ஒப்பிடுதல்	50
10.7 மேலும் சில ஒப்பீடுகள்	51
10.8 ஒன்றுக்கு மேற்பட்ட ஒப்பீடுகள்	52
10.9 இதுவா? அல்லது, அதுவா?	54
10.10 ஒரே பணியைத் திரும்பத் திரும்பச் செய்தல்	55
10.11 நிறுத்துதல், தொடர்தல்.....	58
10.12 அதிக விவரங்கள்	60
10.13 நிரல்பாகங்கள்	62
11.....11 -11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)	63
11.1 பதிப்பி.....	63
11.2 உள்ளீடு.....	63
11.3 ஆனால்.....	64
11.4 இல்லைஆனால் & இல்லை.....	64
11.5. வரை.....	65
11.6. செய் & முடியேனில்.....	66
11.7. ஒவ்வொன்றாக.....	66
11.8. ஆக.....	67
11.9. தேர்ந்தெடு, தேர்வு & ஏதேனில்.....	67
11.10 நிறுத்து.....	68
11.11 தொடர்.....	68
11.12 பட்டியல்.....	69
11.13 பின்இணை.....	69
11.14 எடு.....	69
11.15 தலைகீழ்.....	70
11.16 வரிசைப்படுத்து.....	70

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

11.17 நீளம்.....	70
11.18 நீட்டிக்க.....	70
11.19 dict.....	71
11.20 வை.....	71
11.21 எடு.....	71
11.22 நிரல்பாகம்.....	72
11.23 பின்கொடு.....	72
11.24 கோப்பை_திற.....	72
11.25 கோப்பை_எழுது.....	73
11.26 கோப்பை_முடு.....	73
11.27 கோப்பை_படி.....	74

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

காரணத்திற்காக காலியாக வைக்கப்படும் பக்கம்.

This page was intentionally left blank

1 -வாங்க, நிரல் எழுதுவோம்!

1 - வாங்க, நிரல் எழுதுவோம்!

யாதும் ஊரே, யாவரும் கேளிர்!

கணினி, தரவு (logic) மொழியைத்தவிர எந்த மொழியையும் இயல்பாக அறியாது. அகையால், ஆங்கிலத்தில் மட்டுமன்றி, தமிழிலும் நீங்கள் கணினியை கட்டளையிட எழுதலாம்! இதற்குப் பெயர் நிரலாக்கம்; இப்படி அமைந்த ஒரு மொழி தான் எழில்.

இந்த புத்தகத்தின் வழியாக எழில், மேலும் நிரலாக்கம், இரண்டும் கற்றுக்கொண்டு, Java, Python பொன்ற மொழிகளை கற்க தயாராகிவிடுவீர்கள் என நம்புகிறேன்.

"எழில்" நிரல் மொழி நாம் நன்கு அறிந்த தமிழிலேயே நிரல்கள் எழுத வழிவகை செய்கிறது. ஆனால், நம்முடைய முதல் நிரலை எப்படி எழுதுவது? நேரடியாகக் கணினியில் நிரல் எழுத உட்கார்ந்துவிடலாமா?

1.1 முதல் நிரலை எப்படி எழுதுவது

நாம் ஒரு வீடு கட்டுகிறோம் என்று வைத்துக்கொள்வோம். முதல் வேலையாகச் செங்கலும் சிமெண்டுமாக நிலத்தில் இறங்கிவிடுகிறோமா? அதற்கு முன்னால் பல வேலைகள் உள்ளன:

1. நிலத்தைத் தீர்மானிப்பது
2. அங்கே என்ன கட்டப்போகிறோம் (வீடா, அலுவலகமா, கடையா, வேறு ஏதாவதா) என்று தீர்மானிப்பது
3. அதற்குத் திட்டம் போடுவது (Blueprint)
4. எங்கே சுவர்கள், எங்கே தூண்கள் என்று தீர்மானித்து, திட்டத்துக்கேற்ப அஸ்திவாரம் போடுவது
5. அதன்மீது வீட்டை எழுப்புவது
6. உள்ளே நுழைந்து எல்லாம் திருப்தியாக உள்ளதா என்று பார்ப்பது
7. நண்பர்கள், உறவினர்களுக்குச் சொல்லுவது
8. இதே ஏழு நிலைகள், கணினி நிரல் எழுதுவதிலும் உண்டு:
9. நிலத்தைத் தீர்மானிப்பதுபோல், இங்கே களத்தை, கணினி மொழியைத் தீர்மானிக்கிறோம்
10. எதைக் கட்டப்போகிறோம் என்று தீர்மானிப்பதுபோல், இங்கே என்ன நிரல் எழுதுவது எனத் தீர்மானிக்கிறோம்
11. திட்டம் போடுவதுபோல், இங்கே Algorithm எனப்படும் நிரல் வழிமுறையை எழுதுகிறோம்
12. அஸ்திவாரம் போடுவதுபோல், இங்கே pseudocode எனப்படும் மாதிரி நிரலை எழுதுகிறோம்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

13. வீட்டை எழுப்புவதுபோல், இங்கே முழுமையான நிரலை எழுதுகிறோம்
14. உள்ளே நுழைந்து பார்ப்பதுபோல், இங்கே நாம் எழுதிய நிரலைப் பரிசோதிக்கிறோம் (Testing)
15. நண்பர்கள், உறவினர்களுக்குச் சொல்லுவதுபோல், இங்கே நம்முடைய நிரலை நிரந்தரமாகச் சேமித்துவைத்து வேண்டும்போதெல்லாம் பயன்படுத்துகிறோம்

இப்படி மொத்தமாகச் சொன்னால் குழப்பமாகதான் இருக்கும். அதற்குப் பதிலாக, இந்த ஏழு நிலைகளைப் பின்பற்றி, ஒரு மாதிரி நிரல் எழுதிப்பார்ப்போமா?

1.1 களத்தை, கணினி மொழியைத் தீர்மானிப்பது

இதில் நமக்குக் குழப்பமே வேண்டியதில்லை. "எழில்" மொழியைதான் பின்பற்றப்போகிறோம் என்று நாம் ஏற்கெனவே தீர்மானித்துவிட்டோம்.

1.1.1. ஆனால், அதை எங்கே எழுதுவது?

நம்முடைய விண்டோஸ் அல்லது லைனக்ஸ் வகைக் கணினியில் "எழில்" மொழியில் நிரல் எழுதலாம். ஆனால் அதற்கு நீங்கள் "எழில்" மென்பொருளை அந்தக் கணினியில் நிறுவியிருக்கவேண்டும். ஒருவேளை நீங்கள் இன்னும் அதனைச் செய்யவில்லை என்றால், இணையதளம் <http://www.ezhillang.org> க்கு வாருங்கள். அங்கே நீங்கள் "எழில்" மென்பொருளைத் தரவிறக்கம் செய்யலாம். அல்லது, நேரடியாக அங்கேயே நிரல் எழுதிச் சரி பார்க்கலாம்.

புதிதாக நிரல் எழுதுவோர் பயிற்சிக்காக இணையதளம் <http://www.ezhillang.org> லேயே நிரல் எழுதுவது நல்லது. அதில் நல்ல அனுபவம் ஏற்பட்டபிறகு நம் கணினியில் "எழில்"-ஐ நிறுவிக்கொள்ளலாம்.

1.1.2. என்ன நிரல் எழுதுவது எனத் தீர்மானிப்பது

நீங்கள் உங்கள் விருப்பம்போல் எதற்கும் நிரல் எழுதலாம். இங்கே ஓர் எளிய உதாரணமாக, இரு எண்களுடைய சராசரியைக் கண்டறிவதற்கு நிரல் எழுதுவோம்.

ஆங்கிலத்தில் இதனை Program Objective என்பார்கள். அதாவது, நாம் எழுதப்போகும் நிரலின் நோக்கம் என்ன? அது எப்படிச் செயல்படவேண்டும்? இதைத் தெளிவாகத் தெரிந்துகொண்டால்தான், நாளைக்கே அதில் பிழைகள் ஏதும் இருந்தால் கவனித்துச் சரிப்படுத்தமுடியும்.

ஆக, நம்முடைய நிரலின் நோக்கம், இரண்டு எண்களின் சராசரியைக் கண்டுபிடிப்பது.

1.1.3. நிரல் வழிமுறை (Algorithm)

1 -வாங்க, நிரல் எழுதுவோம்!

நிரல் வழிமுறை என்பது, இந்த நோக்கத்தை நாம் எப்படி நிறைவேற்றப்போகிறோம் என்பதற்கான திட்டமிடல். அதாவது, படிப்படியாகச் சிந்திப்பது.

ஆங்கிலத்தில் **Algorithm** என்று அழைக்கப்படும் நிரல் வழிமுறையை எழுதுவதற்குப் பல வழிகள் உண்டு, அழகழகாக பொம்மை போட்டு எழுதுவதற்கு நிறைய **tools** கூட உண்டு. அதையெல்லாம் நாம் பின்னால் கற்றுக்கொள்வோம். இப்போதைக்கு, ஒரு காகிதத்தில் 1, 2, 3 என்று எண் போட்டு எழுதினால் போதுமானது.

நம் நோக்கம், சராசரி கண்டுபிடிப்பது. அதற்கு உள்ளீடு (Input) என்ன?

இரண்டு எண்கள். நாம் அவற்றை எண்1, எண்2 என அழைப்போம்.

இந்த நிரலின் வெளியீடு (Output or Result) என்ன?

நாம் தந்த இரு எண்களின் சராசரிதான் அது. இந்த எண்ணை நாம் எண்3 என்று அழைப்போம்.

எண்1, எண்2 ஆகியவற்றை வைத்துக்கொண்டு நாம் எப்படி எண்3-ஐக் கண்டறிவது? இதற்கான கணிதச் சூத்திரம் என்ன?

இரு எண்களின் சராசரி என்பது, அவற்றைக் கூட்டி இரண்டால் வகுப்பதுதான். அதாவது:

$$\text{எண்3} = (\text{எண்1} + \text{எண்2}) / 2$$

அவ்வளவுதான். நாம் நமது நிரல் வழிமுறையை எழுதிவிட்டோம். இதோ இப்படி:

உள்ளீடு: எண்1, எண்2

1. எண்1 என்ற எண்ணைப் பெறுக
2. எண்2 என்ற எண்ணைப் பெறுக
3. இவ்விரு எண்களையும் கூட்டுக
4. வந்த கூட்டுத் தொகையை இரண்டால் வகுக்க
5. கிடைத்த விடையை எண்3 எனச் சேமிக்க
6. திரையில் எண்3 என்ற விடையை அச்சிடுக

வெளியீடு: எண்3

இது ஒரு நேரடியான நிரல் வழிமுறை. அதாவது, ஒன்றுக்குப் பிறகு இரண்டு, அதன்பின் மூன்று என வரிசையில் செல்வது. சில நிரல் வழிமுறைகள் அவ்வாறின்றி குதித்துச் செல்லும், திரும்பிச் செல்லும் (உதாரணமாக, 1, 2, 10 அல்லது, 1, 2, 3, 4, 3, 4, 3, 4, 5 என்பதுபோல).

இப்படிப்பட்ட சிக்கலான நிரல் வழிமுறைகளை நாம் பின்னர் தெரிந்துகொள்வோம். இப்போதைக்கு நமது முதல் கணினி நிரலை எழுத இந்த எளிய வழிமுறை போதுமானது.

1.1.4. மாதிரி நிரல் எழுதுவது

ஒரு முக்கியமான கடிதத்தை எழுதுவதற்குமுன்னால் மனத்தில் அதை எழுதிப் பார்க்கிறோம், அல்லது ஒரு துண்டுக் காகிதத்தில் எழுதுகிறோம் அல்லவா? அதுபோல "எழில்" மொழியில் இந்த

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

நிரலை எழுதுவதற்குமுன்னால், மாதிரி நிரல் ஒன்றை எழுதிப் பார்த்துவிடுவோம்.

இங்கே நாம் "எழில்" மொழியின் குறிச்சொற்களைச் சரியாகப் பயன்படுத்தவேண்டும் என்று அவசியம் இல்லை. சாதாரணத் தமிழில் எழுதினாலே போதும். இதோ இப்படி:

```
எண்1 பெறுக
எண்2 பெறுக
கூட்டுத்தொகை = எண்1 + எண்2
எண்3 = கூட்டுத்தொகை / 2
எண்3 அச்சிடுக
```

இது ஒரு மாதிரி நிரல்தான். இதனை நாம் "எழில்" இணையத்தளத்தில் வைத்து இயக்கினால் விடை கிடைக்காது, பிழை(Error)தான் கிடைக்கும்.

அதனால் தவறில்லை. நம் நிரல் எப்படி இருக்கவேண்டும் என்று புரிந்துகொண்டுவிட்டோம் அல்லவா? அதுதான் நம் நோக்கம்.

1.1.5. முழுமையான நிரல் எழுதுவது

இதுதான் நம்முடைய ஏழு படிநிலைகளில் மிக முக்கியமானது. நம்மிடம் உள்ள மாதிரி நிரலை எடுத்துக்கொண்டு, ஒவ்வொரு வரியாக "எழில்" மொழிக்கு ஏற்றபடி அதனை மாற்றப்போகிறோம்.

இதற்குத் தேவையான உதவிக் குறிப்புகள், குறிச்சொற்கள், மாதிரி நிரல்கள் என அனைத்தும் "எழில்" இணையத் தளத்தில் கிடைக்கின்றன. அவற்றின் துணையோடு படிப்படியாக நாம் இதில் முன்னேறவேண்டும். ஒவ்வொரு வரியையும் சரியான "எழில்" மொழிக் கட்டளைகளாக மாற்றவேண்டும்.

உதாரணமாக, "எண்1 பெறுக" என்ற வரி இப்படி மாறும்:

```
எண்1 = 10
```

அடுத்து, "எண்2 பெறுக" என்ற வரி, இதுவும் எளிமையானதுதான்:

```
எண்2 = 6
```

மூன்றாவதாக, கூட்டுத்தொகை கணக்கிடுவது. இந்த வரி மாதிரி நிரலில் உள்ளதுபோலவே "எழில்" மொழியிலும் இயங்கும்:

```
கூட்டுத்தொகை = எண்1 + எண்2
```

நான்காவது வரியும் இதேபோல்தான், மாதிரி நிரலில் உள்ளது அப்படியே இங்கேயும் வரும்:

```
எண்3 = கூட்டுத்தொகை / 2
```

நிறைவாக, விடையை அச்சிடும் ஐந்தாவது வரி இப்படி மாறும்:

```
பதிப்பி "நீங்கள் தந்த எண்களின் சராசரி: ", எண்3
```

அவ்வளவுதான்! வாழ்த்துகள்! நீங்கள் உங்களது முதலாவது "எழில்" நிரலை எழுதிவிட்டீர்கள்:

1 -வாங்க, நிரல் எழுதுவோம்!

```
எண்1 = 10
எண்2 = 6
கூட்டுத்தொகை = எண்1 + எண்2
எண்3 = கூட்டுத்தொகை / 2
பதிப்பி "நீங்கள் தந்த எண்களின் சராசரி: ", எண்3
```

1.1.6. நிரலைப் பரிசோதிப்பது

நாம் நிரல் எழுதிவிட்டோம். ஆனால் அது சரியாக இயங்குகிறதா என்று பார்க்கவேண்டுமே. இதற்கு நாம் "எழில்" மென்பொருளைப் பயன்படுத்தலாம், அல்லது, "எழில்" இணையத் தளத்துக்குச் செல்லலாம். இது நமது முதல் நிரல் என்பதால், அதனை நேரடியாக இணையத் தளத்தில் இயக்கிப் பார்ப்போம்.

உங்கள் இணைய உலாவி(Browser)-ஐத் திறந்து <http://ezhillang.org> என்ற இணையதளம் (website) முகவரிக்குச் செல்லுங்கள். அங்கே "எழில் நிரல் எழுத இங்கே கிளிக் செய்யவும் (Click Here)" என்ற இணைப்பு காணப்படும். அதனை க்ளிக் செய்யுங்கள்.

இப்போது, மேலே நாம் காகிதத்தில் எழுதிய நிரலை அங்கே தட்டச்சு செய்யவேண்டும். உங்களுக்குத் தமிழ் தட்டச்சு செய்வது சிரமம் என்றால், அங்கே இதற்காகத் தரப்பட்டுள்ள விசேஷப் பொத்தான்களையோ தட்டச்சு உதவி நிரலையோ பயன்படுத்துங்கள்.

அடுத்து "நிரலை இயக்குங்கள்" என்ற பொத்தானை அழுத்துங்கள். இந்த விடை திரையில் தோன்றும்:

```
நீங்கள் தந்த எண்களின் சராசரி: , 8
```

வாழ்த்துக்கள்! உங்கள் நிரல் வெற்றிகரமாக இயங்குகிறது.

ஏதேனும் பிழைகள் இருந்தால், பிரச்சனையில்லை, பின்னே சென்று அவற்றைச் சரி செய்து விடையைக் கண்டறியும்வரை நிறுத்தாதீர்கள்.

10, 6 என்ற எண்களுக்குப் பதில், வேறு சில எண்களை இட்டும் நீங்கள் பார்க்கலாம். சராசரி எண் எப்போதும் சரியாக வரவேண்டும்.

1.1.7. நிரலைச் சேமித்துவைப்பது

ஒருமுறை நிரல் எழுதியபிறகு, அதனை நாம் திரும்பத் திரும்பப் பலமுறை இயக்கவேண்டியிருக்கலாம். இதற்காக, நாம் அதனைக் கோப்பாகச் சேமித்துவைப்பது அவசியம்.

"எழில்" மொழியில் எழுதப்படும் நிரல்கள் அனைத்தும் ".n" என்ற நீட்சியுடன் சேமிக்கப்படவேண்டும். உதாரணமாக, இது சராசரி குறித்த நிரல் என்பதால் இதனை "average.n" என்று சேமிக்கலாம்.

இதற்கு நீங்கள் உங்கள் கணினியில் உள்ள notepad போன்ற ஒரு 'text editor'-ஐத் திறக்கவேண்டும். அதற்குள் உங்கள் நிரலைத் தட்டச்சு செய்யலாம், அல்லது மேலே நாம் "எழில்" இணையத் தளத்தில் தட்டச்சு செய்ததைப் பிரதியெடுத்து ஒட்டலாம் (cut-paste).

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

பின்னர், அதனைச் சேமிக்கும்போது, இந்த விவரங்களைத் தரவேண்டும்:

- பெயர்: **average.n**
- சேமிக்கும் இடம்: உங்கள் விருப்பப்படி
- **Encoding: UTF-8**

இதுபோல் நீங்கள் எத்துணை ".n" கோப்புகளை வேண்டுமானாலும் சேமிக்கலாம். பின்னர் வேண்டியபோது அவற்றை எடுத்துப் பயன்படுத்தலாம்.

பயிற்சி:

இதுவரை நாம் பார்த்த அதே ஏழு படிநிலைகளைப் பின்பற்றி, இந்த வகைகளில் நிரல்கள் எழுதிப் பாருங்கள்:

- ஓர் எண்ணின் வர்க்கத்தைக் கண்டறிதல்
- ஓர் எழுத்துச் சரத்தில் எத்தனை எழுத்துகள் உள்ளன என்று கண்டறிதல்
- தரப்படும் மூன்று எண்களில் மிகச் சிறியது எது என்று கண்டறிதல்

இவ்வகை எளிய நிரல்களை எழுதி நன்கு பழகியபிறகு, நீங்கள் இன்னும் சிக்கலான நிரல்களை எழுதத் தொடங்கலாம். அதற்கான உதவிக் குறிப்புகள் அனைத்தையும் "எழில்" இணையத் தளமே உங்களுக்கு வழங்குகிறது.

தொடர்ந்து எழுதுங்கள். வெற்றிக்கு வாழ்த்துகள்!

2 -கணினி மென்பொருள் நிரல் எழுதக் கற்றுக்கொள்

2 - கணினி மென்பொருள் நிரல் எழுதக் கற்றுக்கொள்

அனைவருக்கும் வணக்கம்.

தமிழில் கணினி மென்பொருள் நிரல் (Computer Software Program) எழுதக் கற்றுக்கொள்ளவேண்டும் என்கிற ஆர்வத்துடன் இந்த "எழில்" தளத்துக்கு வந்திருக்கிறீர்கள். உங்களுக்கு எங்கள் வாழ்த்துகள்!

"எழில்"பற்றி மேலும் அறியுமுன்னர், நிரல் என்றால் என்ன, அதைக்கொண்டு நாம் என்ன செய்யமுடியும் என்று ஒரு சுருக்கமான அறிமுகத்தைத் தெரிந்துகொள்வது நல்லது.

ஆங்கிலத்தில் 'கம்ப்யூட்டர்' எனப்படும் கணினி ஓர் அற்புதமான சாதனம். ஆனால், அதற்குச் சுய அறிவு என்று எதுவும் கிடையாது. மனிதர்களாகிய நாம் சொல்வதைத் திரும்பச் செய்யும், அவ்வளவுதான்.

உங்கள் வீட்டில் ஒரு நாய்க்குட்டி இருப்பதாக வைத்துக்கொள்ளுங்கள். அதனிடம் 'உட்கார்' என்று சொன்னால் அது உட்கார்கிறது, 'நில்' என்று சொன்னால் அது நிற்கிறது.

இங்கே 'உட்கார்', 'நில்' என்ற சொற்களை நாம் 'கட்டளை' என்று அழைக்கிறோம். நீங்கள் இடுகிற கட்டளைக்கு ஏற்ப அது செயல்படுகிறது. அவ்வளவுதான்.

கணினியும் அப்படிதான், நீங்கள் தரும் கட்டளைக்கு ஏற்ப அது இயங்கும்.

ஒரே பிரச்னை, 'உட்கார்', 'நில்' போன்ற கட்டளைகள் அதற்குப் புரியாது. தெலுங்கு தெரிந்த ஒருவரிடம் நீங்கள் தமிழைத் தவிர்த்துத் தெலுங்கில் பேசுவதுபோல, கணினிக்குத் தெரிந்த மொழியில் பேசினால்தான் அதற்குப் புரியும்!

கணினிக்கு என்னென்ன மொழிகள் தெரியும்?

ஆங்கிலத்தில் இதற்கு ஏராளமான மொழிகள் உள்ளன. உலகெங்கும் லட்சக்கணக்கானோர் இதனைப் பயன்படுத்திக் கணினிக்குக் கட்டளையிடுகிறார்கள், கணினியும் அதன்படி செயல்படுகிறது.

இந்தக் கட்டளைகளின் தொகுப்புதான், சாஃப்ட்வேர் ப்ரொக்ராம், அல்லது, மென்பொருள் நிரல்.

இப்போது, ஆங்கிலம் அறியாதவர்களும் கணினி நிரல் எழுதக் கற்கவேண்டும் என்ற எண்ணத்துடன், தமிழில் "எழில்" என்ற இந்த மொழிஅறிமுகப்படுத்தப்படுகிறது.

என்ன சொல்கிறீர்கள்? தமிழ் என்பதே ஒரு மொழிதானே, அப்படிப் பார்த்தால் "எழில்" என்பது இன்னொரு மொழியாகதானே இருக்கவேண்டும்?

உண்மைதான், தமிழ் என்கிற, உங்களுக்கு ஏற்கெனவே நன்றாகத் தெரிந்த மொழியில் உள்ள அதே சொற்களைப் பயன்படுத்தி உருவாக்கப்பட்ட இன்னொரு மொழிதான் "எழில்". இதை நீங்கள் காகிதத்திலும் எழுதலாம், கணினியிலும் எழுதலாம். அதன்மூலம் கணினியை உங்கள் விருப்பம்போல் இயக்கலாம்.

இந்தச் சிறு கையேட்டில், "எழில்" மொழியைப் பயன்படுத்துவதற்கான நுட்பங்களை உங்களுக்குச் சொல்லித்தரப்போகிறோம். ஒவ்வொரு படியாக முன்னேறி, மிக விரைவில் "எழில்"மூலம் நினைத்ததையெல்லாம் செய்யத் தெரிந்துகொள்ளப்போகிறீர்கள், கணினி உங்களுடைய அடிமையாகக்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

கை கட்டி நிற்கப்போகிறது.

2.1 "எழில்" மொழியைக் கொண்டு நாம் என்னவெல்லாம் செய்யலாம்?

ஏராளமாகச் செய்யலாம். மொத்தத்தையும் இங்கே சொல்வது சாத்தியமில்லை, அவற்றை நீங்களே பயன்படுத்தித் தெரிந்துகொள்வதுதான் நன்றாக இருக்கும். என்றாலும், ஒரு சிறிய பட்டியல் இங்கே:

1. நீங்கள் விரும்பும் விஷயங்களைத் திரையில் அச்சிடலாம்
2. எளிய, சிக்கலான கணக்குகளைப் போடலாம்
3. தர்க்க அடிப்படையிலான (Logical) தீர்மானங்களை எடுக்கலாம்
4. படம் வரையலாம்
5. ஒரே செயலை நூறு முறை, ஆயிரம் முறை, கோடி முறை திரும்பத் திரும்பச் செய்யலாம்
6. முக்கியமாக, இதன்மூலம் கற்றுக்கொண்ட விஷயங்களை அடிப்படையாக வைத்து, மற்ற கணினி மொழிகள், இதைவிடப் பெரிய, பயனுள்ளவற்றைத் தெரிந்துகொள்ளலாம்

2.2 எழில் அறிமுகம் மற்றும் வரலாறு

உலகெங்கும் கணினி நிரல் எழுதத் தெரிந்தவர்களுக்கு நல்ல வேலை வாய்ப்புகள் உண்டு. ஆனால் அதைவிடப் பெரிய விஷயம், நிரல் எழுதப் பழகுவதன்மூலம் உங்களுக்குத் தர்க்கரீதியிலான சிந்தனை நன்கு பழகும், அது உங்களது படிப்பிலும், மற்ற செயல்பாடுகளிலும் பிரதிபலிக்கும்.

வாருங்கள், "எழில்" உலகத்தினுள் செல்வோம். அது ஒரு பரவசமான பயணம்!

அதற்குமுன்னால், "எழில்" மொழிபற்றிய சில அடிப்படை விஷயங்களைத் தெரிந்துகொண்டுவிடுவது நல்லது.

எழில் என்பது, தமிழில் உருவாக்கப்பட்ட கணினி நிரல் மொழி. மிகவும் எளிமையானது, திறமூலம் (Open Source) அடிப்படையில் வெளியிடப்படுவது.

இதன் நோக்கம், தமிழைத் தாய்மொழியாகக் கொண்ட மாணவர்கள் சுலபமாகக் கணினி நிரல் எழுதக் கற்பிப்பது. இதைக் கொண்டு அவர்கள் தர்க்கரீதியில் சிந்திப்பது, கணக்குகள் போடுவது, கணினியியல் போன்றவற்றை ஆங்கிலத்தின் துணை இன்றியே அறியமுடியும்.

எழில் நிரல் மொழியில், தமிழ்ச் சொற்களும், இலக்கணமும் மிக எளிமையாக அமைக்கப்பட்டுள்ளது. அதேசமயம் மற்ற நவீன கணினி நிரல் மொழிகளில் (ஆங்கிலம் அடிப்படையிலானவை) உள்ள எல்லா வசதிகளும் இதிலும் உண்டு.

பயனுள்ள இந்த எழில் கணினி நிரல் மொழி, இலவசமாகவே வழங்கப்படுகிறது. 2007 ம் ஆண்டுமுதல் உருவாகிவரும் இந்த மொழி, 2009 ம் ஆண்டு முறைப்படி வெளியானது.

எழில் மொழியில் நிரல் எழுதுவதற்கான வழிமுறைகள், கிட்டத்தட்ட BASIC கணினி மொழியைப்போலவே அமைந்திருக்கும். நீங்கள் எழுதும் நிரல்கள் ஒன்றின் ஒன்று என்ற வரிசையில் இயக்கப்படும். அல்லது, Functions எனப்படும் 'நிரல் பாக'ங்களை நீங்கள் பயன்படுத்தலாம்.

இந்த மொழியில் எண்கள், எழுத்துச் சரங்கள், தர்க்கக் குறியீடுகள், பட்டியல்கள் போன்ற வகைகள் உண்டு. நீங்கள் பயன்படுத்தும் மாறி(Variable)களைத் தனியே அறிவிக்க(Declaration)த் தேவையில்லை. நேரடியாக நிரலில் பயன்படுத்தத் தொடங்கலாம்.

2 -கணினி மென்பொருள் நிரல் எழுதக் கற்றுக்கொள்

அதேசமயம், ஒரு வகை மாறியை இன்னொரு வகை மாறியாக மாற்றுவது என்றால், அதற்கு உரிய குறிச்சொற்களைப் பயன்படுத்தவேண்டும்.

2.3 எழில் மொழியின் நோக்கங்கள்:

1. கல்வி : ஒன்றாம் வகுப்பு முதல் 12 ம் வகுப்புவரை பயிலும் மாணவர்கள் இதனைப் பயன்படுத்தி அடிப்படைக் கணினியியல், நிரல் எழுதுதல்பற்றி அறியலாம்
2. இயல்பானது : எழில் மொழியின் இலக்கணம், தமிழின் எழுத்து இலக்கணத்தை ஒட்டியே அமைக்கப்பட்டுள்ளது. ஆகவே, தமிழில் பேசுகிற, எழுதுகிற எவரும் இதனைச் சுலபமாக அறிந்துகொண்டு பின்பற்றலாம்

இதில் உள்ள வசதிகள்:

1. கணிதம் மற்றும் தர்க்கரீதியிலான குறியீடுகள்
2. முந்நூற்றைம்பதுக்கும் மேற்பட்ட நிரல் பாகங்கள் இதில் ஏற்கெனவே சேர்க்கப்பட்டுள்ளன
3. இவைதவிர, நீங்களே புதிய நிரல் பாகங்களை எழுதிச் சேர்க்கலாம்
4. Notepad++ மற்றும் Emacs பயன்படுத்துகிறவர்கள், தங்களது நிரல்களுக்கு ஏற்ற வண்ணக் குறியீடுகளை அமைத்துக்கொள்ளும் வசதி எழில் மொழியில் உண்டு

2.4 குறிச்சொற்கள்:

எழில் மொழியில் நிரல் எழுதுவதற்கு ஏராளமான குறிச் சொற்கள் உள்ளன. இவற்றை நாம் அடுத்தடுத்த அத்தியாயங்களில் விளக்கமாகப் பார்க்கவிருக்கிறோம்.

இப்போதைக்கு, எழில் மொழியில் உள்ள முக்கியமான குறிச் சொற்களின் பட்டியலையும், அதற்கு இணையான ஆங்கிலச் சொற்களையும் இங்கே பார்த்துவிடுவோம்:

```
ஆக : for
நிறுத்து : break
தொடர் : continue
பின்கொடு : return
ஆனால் : if
இல்லைஆனால் : elseif
இல்லை : else
தேர்ந்தெடு : select
தேர்வு : case
ஏதேனில் : otherwise
வரை: while
செய் : do
முடியேனில் : until
பதிப்பி : print
நிரல்பாகம் : function
முடி : end
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

3 -எழில் பதிவிறக்கம் மற்றும் நிறுவ (Installing Ezhil)

3 - எழில் பதிவிறக்கம் மற்றும் நிறுவ (Installing Ezhil)

இப்போது, "எழில்" மொழியைக் கொண்டு தமிழிலேயே கணினி நிரல், அதாவது Software Program எப்படி எழுதுவது என்று கற்றுக்கொள்வோம்.

அதற்குமுன்னால், நீங்கள் ஏற்கெனவே "எழில்" மொழியைப் பயன்படுத்துவதற்கான மென்பொருளை (Software) இணையத்திலிருந்து பதிவிறக்கம் செய்துவிட்டீர்கள் (Downloaded), உங்கள் கணினியில் அதை நிறுவிவிட்டீர்கள் (Installed) என நம்புகிறேன். ஒருவேளை நீங்கள் அதைச் செய்யவில்லை என்றால், உடனடியாக <http://ezhillang.org/> என்ற இணைய தளத்துக்குச் செல்லுங்கள், அங்கே தரப்பட்டுள்ள உதவிக் குறிப்புகளின் அடிப்படையில் உங்கள் கணினியை "எழில்" மயமாக்குங்கள்!

அடுத்து, நாம் நிரல் எழுதத் தொடங்கவேண்டும். ஆனால், எங்கே?

இதற்கு நாம் Interpreter என்ற பகுதியைத் தொடங்கவேண்டும். அதாவது, உங்களது நிரலை வாசித்து, அதைச் செயல்படுத்தி விடைகளைத் திரையில் காட்டுகிற பகுதி. நம் வசதிக்காக, இதனை "நிரல் மேசை" என அழைப்போம்!

உங்களது எழில் கணினித் திரையில் "ez" என்று தட்டச்சு செய்யுங்கள். விசைப்பலகையில் உள்ள "Enter" விசையைத் தட்டுங்கள்.

உடனே, உங்கள் திரையில் Interpreter, அதாவது நிரல் மேசை தோன்றும். இதில்தான் நாம் நம்முடைய நிரல்களை எழுதப்போகிறோம்***

3.1 வணக்கம் தமிழகம்!

இப்போது, ஒரு மிக எளிய நிரல் எழுதுவோம். இதன் பெயர், 'வணக்கம்'.

அதாவது, உங்கள் கணினி உங்களுக்கு வணக்கம் சொல்லப்போகிறது. அதற்கு நீங்கள் பிறப்பிக்கவேண்டிய கட்டளை இது:

பதிப்பி "வணக்கம் தமிழகம்!"

இந்தக் கட்டளையை நீங்கள் தட்டச்சு செய்து, "Enter" என்ற விசையைத் தட்டுங்கள்.

மறுவிநாடி, உங்கள் திரையில் "வணக்கம் தமிழகம்!" என்ற எழுத்துகள் தோன்றும்.

வாழ்த்துக்கள்! நீங்கள் உங்களுடைய முதல் நிரலை எழுதிவிட்டீர்கள்!

இது மிக எளிமையான நிரல்தான். இன்னும் சிக்கலான, பயனுள்ள பல செயல்களைச் செய்யக்கூடிய நிரல்களை அடுத்தடுத்து எழுதப்போகிறீர்கள்.

அதற்குமுன்னால், நாம் தொடங்கிய நிரல் மேசையை எப்படி மூடிவைப்பது என்று தெரிந்துகொள்ளவேண்டும். அதற்கு நீங்கள் பயன்படுத்தவேண்டிய கட்டளைச் சொல் exit()

இதைத் தட்டச்சு செய்து "Enter" பொத்தானை அழுத்தியதும், நிரல் மேசைத் திரை காணாமல் போய்விடும்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

பிறகு, நமக்கு வேண்டியபோது **ez** என்று தட்டச்சு செய்து மீண்டும் "Enter" பொத்தானை அழுத்தினால் போதும், மீண்டும் நிரல் மேசை தோன்றும், அடுத்த நிரலை எழுதத் தொடங்கலாம். இப்படி ஒவ்வொருமுறையும் நிரல் மேசையைத் தொடங்கி, மீண்டும் மூடிவைப்பது சிரமமாக இருந்தால்,

3.2 இணையத்தை பயன்படுத்தும் எழில்

இன்னோர் எளிய வழி இருக்கிறது, நீங்கள் வழக்கமாக இணையத்தை அணுகப் பயன்படுத்தும் **Internet Browser** ல் 'எழில்' மொழி நிரல்களை எழுதலாம். இதோ இப்படி:

1. முதலில், முன்புபோலவே நிரல் மேசையைத் தொடங்கிக்கொள்ளுங்கள்
2. அதில் இப்படி தட்டச்சு செய்யுங்கள்:

```
$ python ezhil/EZWeb.py
```

3. சில விநாடிகள் கழித்து, உங்கள் திரையில் '**Server Starts - localhost:8080**' என்ற வாசகம் தோன்றும்
4. இப்போது, உங்கள் கணினியில் 'எழில்' மொழி நிறுவப்பட்டிருக்கும் இடத்துக்கு (**Folder**) செல்லுங்கள், அங்கே '**Web**' என்ற பகுதியைத் திறங்கள்
5. அதற்குள் '**ezhil_on_the_web.html**' என்ற கோப்பு இருக்கும், அதனைத் திறங்கள்
6. அவ்வளவுதான், இனிமேல் நீங்கள் உங்கள் **Internet Browser** ன் திரையிலேயே 'எழில்' நிரல்களை எழுதலாம், இயக்கலாம், விடையைப் பார்க்கலாம்!

ஒரு விஷயம், **Browser** மூலம் இயங்கும் இந்த "எழில்" வடிவத்தில் சில குறிப்பிட்ட வசதிகள் இல்லை, உதாரணமாக, கணினியைப் பயன்படுத்துகிறவரிடம் ஓர் எண்ணையோ, எழுத்தையோ கோரிப் பெறமுடியாது, படம் வரையமுடியாது.

ஆகவே, இந்தப் புத்தகத்தில் உள்ள சில நிரல்களை நீங்கள் அதில் இயக்கமுடியாமல் போகலாம், அப்போது, நாம் முன்பு பார்த்துள்ள நிரல் மேசையைப் பயன்படுத்திக்கொள்ளுங்கள்.

3.3 எழில் - பைத்தான் வழியில்

பைத்தான் வழியில் எழில் பதிவிறக்கம் மற்றும் நிறுவ

```
$ pip install ezhil
```

```
$ python
```

பிறகு,

```
>>> import ezhil  
>>> ezhil.start()
```

கட்டளையிட்டால் நீங்கள் பைத்தான் வழியில் எழில் செயல்படுத்தலாம்.

3 -எழில் பதிவிறக்கம் மற்றும் நிறுவ (Installing Ezhil)

```
$ python
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import ezhil
>>> ezhil.start()
எழில் - ஒரு தமிழ் நிரலாக்க மொழி (Tue Jul 2 20:22:25 EDT 2013)
Ezhil : A Tamil Programming Language - version 0.76, (C) 2007-2013
Type "help", "copyright", "credits" or "license" for more information.
எழில் 1>>
```

3.4 சில எளிய எழில் கணக்குகள்

இப்போது, "எழில்" மொழியைப் பயன்படுத்திச் சில எளிய கணக்குகளைப் போடுவோம்.

முதலில், நாம் ஏற்கெனவே பார்த்த முறைப்படி உங்கள் நிரல் மேசையைத் திறந்துகொள்ளுங்கள். அதில் கீழே உள்ள வரியைத் தட்டச்சு செய்து "Enter" பொத்தானை அழுத்துங்கள்

பதிப்பி 10+15

இப்போது உங்கள் திரையில் 25 என்ற எண் தோன்றும். அதாவது, 10, 15 ஆகியவற்றின் கூட்டுத் தொகை. இதேபோல், நாம் கழித்தல், பெருக்கல், வகுத்தல் ஆகியவற்றையும் செய்யலாம். இப்படி:

பதிப்பி 30-5

பதிப்பி 5*5

பதிப்பி 25/5

இன்னும் கொஞ்சம் பெரிய கணக்கு, இருபத்தைந்தின் அடுக்கு இரண்டு, அதாவது 25 * 25 எப்படிச் கணக்கிடுவீர்கள்? இதோ, இப்படிதான்:

பதிப்பி 25^2

இந்தக் கணக்குகளை நாம் தனித்தனியே போடவேண்டும் என்று அவசியம் இல்லை, பலவற்றைத் தொகுத்து ஒரே வரியில்கூட எழுதலாம். இப்படி:

பதிப்பி ((5+(25^2))/6)^2)

இங்கே நீங்கள் அடைப்புக்குறி தந்துள்ள வரிசைப்படி "எழில்" ஒவ்வொரு கணக்காகப் போடும், நிறைவாக சரியான விடையைச் சொல்லிவிடும்.

உங்களுடைய கணக்கு வகுப்பில் "பை" என்ற முக்கியமான எண்ணைப்பற்றிப் படித்திருப்பீர்கள். 3.14 என்று தொடங்கும் அந்தப் பயனுள்ள எண்ணை உங்களது "எழில்" கணக்குகளில் பயன்படுத்தவேண்டுமானால், இப்படி எழுதவேண்டும்:

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

பதிப்பி pi()

இதேபோல், Sin, Cos, Tan, Log போன்றவற்றையும் "எழில்" கணக்குகளில் கொண்டுவரலாம்.
இப்படி:

```
radian = pi()/180  
பதிப்பி sin(30*radian)  
பதிப்பி tan(60*radian)  
பதிப்பி log(100*radian)
```

ஆனால், இவையெல்லாம் நாமே தட்டச்சு செய்கிற கணக்குகள். கொஞ்சம் வித்தியாசமாக, நாம் கேட்கிற கணக்குகளைக் கணினியே போட்டுப் பதில் தரும்படி மாற்றமுடியாதா?

நிச்சயமாகச் செய்யலாம், அதுதான் நம்முடைய அடுத்த பணி!

இதுவரை, நாம் தட்டச்சு செய்த விஷயங்களை "எழில்" மொழி அப்படியே நிறைவேற்றிக்கொண்டிருந்தது. இப்போது, அது கேட்கும் கேள்விக்கு நாம் பதில் சொல்லப்போகிறோம்.

இதற்கு நாம் "உள்ளீடு" என்ற கட்டளையைப் பயன்படுத்தவேண்டும். இதுபோல:

```
முதல்எண் = உள்ளீடு("ஓர் எண்ணைச் சொல்லுங்கள்: ")
```

இந்தக் கட்டளையை நிறைவேற்றினால், உங்கள் கணினித் திரையில் "ஓர் எண்ணைச் சொல்லுங்கள்: " என்ற எழுத்துகள் தோன்றும். நீங்கள் ஏதாவது ஓர் எண்ணைத் தட்டச்சு செய்து "Enter" பொத்தானை அழுத்தவேண்டும்.

உடனே, நீங்கள் தட்டச்சு செய்த எண் கணினியால் பெறப்பட்டு, "முதல்எண்" என்ற பெயரில் சேமிக்கப்பட்டுவிடும்.

இப்போது, இரண்டாவதாக இன்னொரு எண்ணையும் உள்ளீடு செய்வோம்.

இப்படி:

```
இரண்டாம்எண் = உள்ளீடு("இன்னொரு எண்ணைச் சொல்லுங்கள்: ")
```

இந்தக் கட்டளை நிறைவேறியதும், நீங்கள் இன்னொரு எண்ணைத் தட்டச்சு செய்யவேண்டும். அது "இரண்டாம்எண்" என்ற பெயரில் சேமிக்கப்பட்டுவிடும்.

இப்போது, இந்த இரண்டு எண்களையும் கூட்டி, "மூன்றாம்எண்" என்ற பெயரில் சேமிப்போம். இது உங்களுக்கு ஏற்கெனவே நன்றாகத் தெரிந்த விஷயம்தானே:

```
மூன்றாம்எண் = முதல்எண் + இரண்டாம்எண்
```

நிறைவாக, இந்த விடையை அச்சிடுவோம், இதோ இப்படி:

3 -எழில் பதிவிறக்கம் மற்றும் நிறுவ (Installing Ezhil)

பதிப்பி "நீங்கள் தந்த எண்களின் கூட்டுத் தொகை: " மூன்றாம்எண்

அவ்வளவுதான். இப்போது, இந்த நிரலை இயக்கிப் பார்ப்போம்.

See : simple_add.n

ஓர் எண்ணைச் சொல்லுங்கள்: 6
இன்னோர் எண்ணைச் சொல்லுங்கள்: 7
நீங்கள் தந்த எண்களின் கூட்டுத் தொகை: , 13

ஓர் எண்ணைச் சொல்லுங்கள்: 10
இன்னோர் எண்ணைச் சொல்லுங்கள்: 5
நீங்கள் தந்த எண்களின் கூட்டுத் தொகை: 15

அருமை! "எழில்" மொழியைப் பயன்படுத்தி ஓர் உபயோகமான நிரல் எழுதிவிட்டீர்கள்!

இப்போது, உங்களுக்கு ஒரு பயிற்சி.

ஒரு கடையில் மக்கள் வந்து காய்கறி, பழம், ரொட்டி வாங்கிச் செல்கிறார்கள். இவற்றின் விலைகளை நீங்கள் தனித்தனியே பெற்று, மொத்தக் கூட்டுத் தொகை எவ்வளவு என்று சொல்லவேண்டும். பின்னர், அவர்கள் தரும் தொகை எவ்வளவு என்று பார்த்து, மீதி சில்லறை எவ்வளவு என்றும் சொல்லவேண்டும்.

இதுவரை நீங்கள் கற்றுக்கொண்ட விஷயங்களை வைத்து அதற்கு ஒரு நிரல் எழுதுங்கள் பார்க்கலாம்!

விடை:

காய்கறி = உள்ளீடு ("காய்கறி வாங்கிய தொகை எவ்வளவு? ")
பழம் = உள்ளீடு ("பழம் வாங்கிய தொகை எவ்வளவு? ")
ரொட்டி = உள்ளீடு ("ரொட்டி வாங்கிய தொகை எவ்வளவு? ")

மொத்தத்தொகை = காய்கறி + பழம் + ரொட்டி

பதிப்பி "நீங்கள் தரவேண்டிய ரூபாய் " மொத்தத்தொகை

தரும்தொகை = உள்ளீடு ("நீங்கள் தரும் தொகை எவ்வளவு? ")

மீதித்தொகை = தரும்தொகை - மொத்தத்தொகை

பதிப்பி "நன்றி. மீதி சில்லறையைப் பெற்றுக்கொள்ளுங்கள்: ரூபாய் " மீதித்தொகை

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

4 - சரங்கள் மற்றும் எழுத்துகள் (Strings)

சென்ற அத்தியாயத்தில் "எழில்" மொழியைப் பயன்படுத்தி எண்களோடு விளையாடினோம், இப்போது எழுத்துகள், வார்த்தைகள், வாக்கியங்களுடன் விளையாடப் பழகுவோம்.

"க" என்பது ஒரு தனி எழுத்து, "ல்" என்பது இன்னொரு தனி எழுத்து, அவை சேர்ந்து "கல்" என்ற வார்த்தை உருவாகிறது. இப்படிப் பல வார்த்தைகள் சேர்ந்து ஒரு வாக்கியமாகும், இப்படி: "கல்லும் மண்ணும் சேர்ந்து கட்டிய வீடு."

இவை அனைத்தையும் நாம் "சரங்கள்" என்று அழைக்கலாம். ஒரு சரம் என்பது தனி எழுத்தாகவோ, வார்த்தையாகவோ, வாக்கியமாகவோ இருக்கலாம்.

உதாரணமாக, "நீ" என்பது ஒரு சரம், "நீர்" என்பது இன்னொரு சரம், "நீர் தருகிறேன்" என்பது இன்னொரு சரம். இப்படி சொல்லிக்கொண்டே போகலாம்.

4.1 சரம் நீளம் (String length)

எழில் மொழியில், இந்தச் சரங்களை வைத்து என்னவெல்லாம் செய்யமுடியும்?

முதலில், ஒரு சரத்தின் நீளத்தைக் கண்டுபிடிப்போம். அதற்கு ஓர் எளிய நிரல் எழுதுவோம்.

```
அ = உள்ளீடு("உங்களுக்குப் பிடித்த ஏதாவது ஓர் எழுத்துச் சரத்தைத் தாருங்கள்: ")
```

```
பதிப்பி "நீங்கள் தந்த சரம்: " அ
பதிப்பி "அந்தச் சரத்தின் நீளம்: " நீளம்(அ)
```

இந்த நிரலை மேலோட்டமாகப் படித்தாலே உங்களுக்கு நன்றாகப் புரியும். நீங்கள் தரப்போகும் கணினியை "எழில்" மொழி வாங்கி, "அ" என்ற பெயரில் சேமிக்கிறது, பின்னர் அதன் நீளத்தைக் கணக்கிட்டு அச்சிடுகிறது. (See demo program: [string_demo.n](#))

இப்போது, நாம் எழுதிய நிரலை இயக்கிப் பார்ப்போம்:

```
உங்களுக்குப் பிடித்த ஏதாவது ஓர் எழுத்துச் சரத்தைத் தாருங்கள்: என் பெயர் எழில்
நீங்கள் தந்த சரம்: என் பெயர் எழில்
அந்தச் சரத்தின் நீளம்: 10
```

4.2 சரம் இணைவதற்கு (String length)

அடுத்து, இரண்டு எழுத்துச் சரங்களை ஒட்டவைப்பது எப்படி என்று பார்க்கலாம்.

இதற்கு, நீங்கள் எண்களைக் கூட்டுவதற்குப் பயன்படுத்திய "+" என்ற குறியீட்டைப் பயன்படுத்தலாம். இதோ இப்படி:

4 - சரங்கள் மற்றும் எழுத்துகள் (Strings)

```
அ = "தமிழ்"  
ஆ = "அமுது"  
இ = அ + ஆ
```

```
பதிப்பி அ  
பதிப்பி ஆ  
பதிப்பி இ
```

இந்த நிரல், "தமிழ்" மற்றும் "அமுது" என்ற இரு எழுத்துச் சரங்களை ஒட்டவைத்துக் காண்பிக்கிறது. அதனை இயக்கினால், "தமிழ்-அமுது" என்று அச்சிடும்.

இப்போது, உங்களுக்கு ஒரு பயிற்சி, "தமிழ்-அமுது" என்ற சொல் பார்ப்பதற்கு நன்றாக இல்லையே, இந்த நிரலைக் கொஞ்சம் மாற்றி, இரு சொற்களுக்கும் நடுவில் ஓர் இடைவெளி விட்டு "தமிழ் அமுது" என்று அச்சிடும்படி செய்யுங்கள், பார்க்கலாம்!

விடை:

```
அ = "தமிழ்"  
ஆ = "  
இ = "அமுது"  
ஈ = அ + ஆ + இ
```

```
பதிப்பி ஈ
```

அடுத்த பயிற்சி. இதுவரை நீங்கள் கற்றுக்கொண்ட விஷயங்களையெல்லாம் வைத்து, கீழே தரப்பட்டுள்ள தேவைக்கேற்ப ஒரு நிரல் எழுதுங்கள்:

- * உங்கள் கணினி இரண்டு எழுத்துச் சரங்களைக் கேட்டுப் பெறவேண்டும்
- * அந்த இரண்டையும் ஒட்டவைத்து மூன்றாவதாக ஒரு சரத்தை உருவாக்கவேண்டும்
- * இந்த மூன்றாவது சரத்தின் நீளத்தைத் திரையில் அச்சிடவேண்டும்

விடை:

```
முதல்சரம் = உள்ளீடு("ஒரு வாக்கியம் சொல்லுங்கள்: ")  
இரண்டாம்சரம் = உள்ளீடு("இன்னொரு வாக்கியம் சொல்லுங்கள்: ")
```

```
மூன்றாம்சரம் = முதல்சரம் + இரண்டாம்சரம்
```

```
சரநீளம் = நீளம்(மூன்றாம்சரம்)
```

```
பதிப்பி "நீங்கள் தந்த இரு சரங்களில் உள்ள மொத்த எழுத்துகள் " சரநீளம்
```

அடுத்து, ஒரு ஜாலியான விளையாட்டு, ஒரு சரத்துக்குள் நாம் விரும்புகிற எழுத்துகள் எங்கே இருக்கின்றன என்று கொஞ்சம் தேடிக் கண்டுபிடிக்கலாமா?

இதற்கு, "சரம்_கண்டுபிடி" என்ற கட்டளை வார்த்தையைப் பயன்படுத்தவேண்டும். இதோ இப்படி:

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

முக்கனிகள் = "மா, பலா, வாழை"

பதிப்பி "முக்கனிகள் : " முக்கனிகள்

பதிப்பி சரம்_கண்டுபிடி(முக்கனிகள், "வாழை")

பதிப்பி சரம்_கண்டுபிடி(முக்கனிகள், "தேங்காய்")

மா, பலா, வாழை என்ற முக்கனிகளில், வாழை உள்ளது, ஆகவே, வாழையைத் தேடும்போது பூஜ்ஜியத்துக்கு மேற்பட்ட ஓர் எண் விடையாகக் கிடைக்கும்.

ஆனால், அந்த முக்கனிகளின் பட்டியலில் தேங்காய் இல்லை. ஆகவே, அதனைத் தேடும்போது பூஜ்ஜியத்தைவிடக் குறைவான (-1) விடை உங்களுக்குக் கிடைக்கும். இதுவே "சரம்_கண்டுபிடி" கட்டளையின் பயன்.

4.3 தேடல், சரங்களை மாற்றவும் (Search Strings)

அடுத்து, முக்கனிகளைக் கொஞ்சம் மாற்றம் செய்வோமா? எனக்கு வாழை பிடிக்காது, அதற்குப் பதிலாகக் கொய்யாவைச் சேர்த்து விளையாடுவோம். இதோ இப்படி:

முக்கனிகள் = "மா, பலா, வாழை"

பதிப்பி "பழைய முக்கனிகள் : " முக்கனிகள்

முக்கனிகள் = சரம்_இடமாற்று(முக்கனிகள், "வாழை", "கொய்யா")

பதிப்பி "புதிய முக்கனிகள் : " முக்கனிகள்

இங்கே "சரம்_இடமாற்று" என்ற கட்டளை "வாழை"யைத் தூக்கி எறிந்துவிட்டு, அந்த இடத்தில் "கொய்யா"வை உள்ளே நுழைத்துவிட்டது. இதனை இயக்கினால், கீழ்க்காணும் விடை கிடைக்கும்:

பழைய முக்கனிகள் : மா, பலா, வாழை

புதிய முக்கனிகள் : மா, பலா, கொய்யா

இப்படி ஒவ்வொரு பழமாகப் பார்த்துப் பார்த்து மாற்றிக்கொண்டிருப்பது சிரமம். இதையே இன்னும் சுலபமாகச் செய்ய ஏதேனும் வழி உண்டா?

அதற்கு நீங்கள் "பட்டியல்" என்ற கட்டளையைப் பயன்படுத்தவேண்டும்.

4.4 பட்டியல்கள் பயன்படுத்த, வரிசைப்படுத்த (Using Lists, Sorting)

வகுப்பில் உங்கள் ஆசிரியர் சொல்லும் விஷயங்களை ஒன்றன்கீழ் ஒன்றாகப் பட்டியல் போடுகிறீர்கள் அல்லவா? அதுபோல, பழங்களை வரிசையாக அடுக்கிவைக்கப் பட்டியலைப் பயன்படுத்தலாம்.

முதலில், ஒரு காலிப் பட்டியலைத் தயாரிப்போம். இதோ, இப்படி:

பழங்கள் = பட்டியல்()

4 - சரங்கள் மற்றும் எழுத்துகள் (Strings)

அடுத்து, இந்தப் பட்டியலில் பழங்களைச் சேர்ப்போம், அதற்கு "பின்இணை" என்ற கட்டளையைப் பயன்படுத்தவேண்டும்:

```
பின்இணை(பழங்கள், "மா")
பின்இணை(பழங்கள், "பலா")
பின்இணை(பழங்கள், "வாழை")
பின்இணை(பழங்கள், "கொய்யா")
பின்இணை(பழங்கள், "ஆப்பிள்")
```

ஆக, இப்போது நம் பட்டியலில் ஐந்து பழங்கள் உள்ளன. அவற்றுக்குத் தனித்தனியே எண்கள் தருவோமா?

ஒரு விஷயம், பட்டியலின் எண்கள் 1, 2, 3 என அமையாது, பூஜ்ஜியத்திலிருந்து தொடங்கும், அதாவது, 0, 1, 2... இப்படி.

```
0 மா
1 பலா
2 வாழை
3 கொய்யா
4 ஆப்பிள்
```

இப்போது, நாம் ஒரு குறிப்பிட்ட பழத்தை அச்சிடவேண்டுமென்றால், "எடு" என்ற கட்டளையைப் பயன்படுத்தவேண்டும். உதாரணமாக, இந்த வரியைப் பாருங்கள்:

```
*****
எடு(பழங்கள், 1)
*****
```

இந்த நிரல் இயங்கினால், "பலா" என்று அச்சிடும். காரணம், அந்தப் பட்டியலில் 1 என்ற எண்ணுக்கு நேராக அந்தப் பழம்தான் பதிவாகியுள்ளது.

அடுத்தபடியாக, இந்தப் பட்டியலில் உள்ள பழங்களை அகர வரிசைப்படி அடுக்குவோமா? அதற்கு நாம் "வரிசைப்படுத்து" என்ற கட்டளையைப் பயன்படுத்தவேண்டும். இதோ, இப்படி:

```
வரிசைப்படுத்து(பழங்கள்)
```

இந்தக் கட்டளை, பழங்களை அகர வரிசைப்படி மாற்றும். அதாவது, முதலில் ஆப்பிள், அடுத்து கொய்யா, பலா, மா, அதன்பிறகு வாழை.

இப்போது, முன்பு நாம் இயக்கிய பழைய கட்டளையை மீண்டும் இயக்குவோம்:

```
எடு(பழங்கள், 1)
```

ஆனால் விடை, பலா இல்லை, கொய்யா! காரணம், நாம் பழங்களை அகர வரிசைப்படி அடுக்கியதுதான்!

இப்போது, உங்களுக்கு ஒரு பயிற்சி, மூன்று நண்பர்களுடைய பெயரைப் பட்டியலில் சேர்த்து, அகர

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

வரிசைப்படி அவர்கள் ஒவ்வொருவருக்கும் வணக்கம் சொல்லி அச்சிடுங்கள். உதாரணமாக, வணக்கம் கணேஷ், வணக்கம் ரமேஷ்... இப்படி.

விடை:

நண்பர்கள் = பட்டியல் ()

பின்இணை(நண்பர்கள், "கணேஷ்")

பின்இணை(நண்பர்கள், "ரமேஷ்")

பின்இணை(நண்பர்கள், "மகேஷ்")

வரிசைப்படுத்து(நண்பர்கள்)

அ = "வணக்கம் "

பதிப்பி அ + எடு(நண்பர்கள், 0)

பதிப்பி அ + எடு(நண்பர்கள், 1)

பதிப்பி அ + எடு(நண்பர்கள், 2)

5 -நிபந்தனை (Conditionals)

5 - நிபந்தனை (Conditionals)

இதுவரை, நாம் எழுதிய நிரல் வகைகள் அனைத்தும், நேர் கோட்டில் செல்பவை. அதாவது, ஒரு வரிக்குப்பிறகு அடுத்த வரி, பின்னர் இன்னொரு வரி... இப்படி.

அவ்வாறில்லாமல், சில காரணிகளுக்கேற்ப ஒரே நிரலை வெவ்வேறுவிதமாக மாறி இயங்கச் செய்ய முடியுமா?

தாராளமாகச் செய்யலாம், இதற்கு நாம் தர்க்கக் குறியீடுகளைப் பயன்படுத்தவேண்டும்.

உதாரணமாக, நீங்கள் வீட்டுப்பாடம் எழுதி முடித்தால், விளையாடச் செல்லலாம் என்று உங்கள் தாய் கூறுகிறார். இதனைத் தர்க்கக் குறியீடுகளின்படி எப்படி எழுதுவது?

- வீட்டுப்பாடம் எழுதி முடித்துவிட்டாயா?
- ஆம் எனில் ... விளையாடலாம்
- இல்லை எனில் ... விளையாடக்கூடாது

5.1 ஆம் இல்லை எனில்

இதே விஷயத்தை "எழில்" மொழியில் ஒரு நிரலாக எழுதிப் பார்ப்போமா? இதோ, இப்படி:

```
அ = உள்ளீடு("வீட்டுப்பாடம் எழுதிவிட்டாயா? ")
```

```
if (அ == "ஆமாம்") ஆனால்
```

```
பதிப்பி "விளையாடச் செல்லலாம்"
```

```
இல்லை
```

```
பதிப்பி "வீட்டுப்பாடம் எழுதியபிறகுதான் விளையாடவேண்டும்!"
```

```
முடி
```

இங்கே நாம் பயன்படுத்தியிருக்கும் தர்க்கக் குறியீடு, இப்படி அமைகிறது:

1. முதலில் "if" என்ற சிறப்பு எழுத்து
2. அடுத்து, அடைப்புக்குறிக்குள் நாம் ஒப்பிடவிரும்பும் விஷயம் (அ = "ஆமாம்")
3. அடுத்து, "ஆனால்" என்கிற சொல்
4. அடுத்த வரியில், அந்த ஒப்பீடு உண்மை எனில், நாம் செய்ய விரும்பும் செயல், இங்கே நாம் ஒன்று அல்லது அதற்கு மேற்பட்ட செயல்களை வெவ்வேறு வரிகளில் தரலாம்
5. அடுத்து "இல்லை" என்கிற சொல்
6. அடுத்த வரியில், அந்த ஒப்பீடு பொய் எனில், நாம் செய்ய விரும்பும் செயல், இங்கேயும் நாம் ஒன்று அல்லது அதற்கு மேற்பட்ட செயல்களை வெவ்வேறு வரிகளில் தரலாம்
7. நிறைவாக "முடி" என்கிற சொல், நாம் செய்யும் தர்க்கரீதியிலான பணிகளை நிறைவு செய்கிறது

இதைப் புரிந்துகொள்வதற்கு, இன்னோர் எளிய உதாரணம் பார்ப்போம்: (see: program - age.n)

```
வயது = உள்ளீடு("உங்கள் வயது என்ன? ")
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

@ (வயது > 18) ஆனால்

பதிப்பி "நீங்கள் தேர்தலில் வாக்களிக்கலாம்"

பதிப்பி "நீங்கள் வாகனம் ஓட்டலாம்"

இல்லை

பதிப்பி "நீங்கள் தேர்தலில் வாக்களிக்க இயலாது"

பதிப்பி "நீங்கள் வாகனம் ஓட்ட இயலாது"

முடி

இதையே கொஞ்சம் விரிவுபடுத்தி, ஒரு சிறிய விநாடி வினா நிகழ்ச்சி நடத்துவோமா?

பதிப்பி "இந்தியாவின் தலைநகரம் எது?"

பதிப்பி "அ. சென்னை"

பதிப்பி "ஆ. மதுரை"

பதிப்பி "இ. டெல்லி"

பதிப்பி "ஈ. மும்பை"

சரியானவிடை = "இ"

உங்கள்விடை = உள்ளீடு ("உங்களுடைய பதில் என்ன? ")

@ (சரியானவிடை == உங்கள்விடை) ஆனால்

பதிப்பி "வாழ்த்துகள். உங்களுக்கு 10 மதிப்பெண்கள்!"

இல்லை

பதிப்பி "தவறான விடை. மீண்டும் முயற்சி செய்யுங்கள்!"

முடி

சில நேரங்களில், ஒரு தர்க்க ஒப்பீட்டுக்குள் இன்னொரு தர்க்க ஒப்பீடும் வரலாம், அப்போது @, ஆனால், இல்லை, முடி ஆகிய நான்கு சொற்களையும் முழுமையாக, அடுத்தடுத்து வரும் ஒரே தொகுப்பாக (Block) ஒன்றுக்குள் இன்னொன்றை அமைத்துப் பயன்படுத்தவேண்டும். கிட்டத்தட்ட உங்கள் கணிதப் பாடத்தில் வரும் அடைப்புக்குறிகளைப்போல என்று வைத்துக்கொள்ளுங்களேன்!

உதாரணமாக, இந்த நிரலைப் பாருங்கள்:

அ = உள்ளீடு ("ஓர் எண்ணைச் சொல்லுங்கள் ")

ஆ = உள்ளீடு ("இன்னொரு எண்ணைச் சொல்லுங்கள் ")

@ (அ == ஆ) ஆனால்

பதிப்பி "இரு எண்களும் சமம்!"

இல்லை

@ (அ > ஆ) ஆனால்

பதிப்பி "முதல் எண் பெரியது!"

இல்லை

பதிப்பி "இரண்டாம் எண் பெரியது!"

முடி

முடி

இதுவரை நாம் பார்த்த நிரல்களோடு ஒப்பிடுகையில், இது கொஞ்சம் பெரியதாக உள்ளது. பின்னர் இன்னொருநாள் வேறொருவரோ (அல்லது நாமேகூட) இதனைப் படித்தால் நமக்குப் புரியுமோ, புரியாதோ!

5 -நிபந்தனை (Conditionals)

ஆகவே, நிரல் எழுதும்போதே, அதற்கான விளக்கங்களையும் ஆங்காங்கே எழுதிவிடுவது நல்லது. இதற்கு நீங்கள் "##" என்ற சின்னத்தைப் பயன்படுத்தவேண்டும். இதோ இப்படி:

```
## இந்த நிரலை எழுதியவர்: சந்திரசேகரன்

## முதல் எண்ணை உள்ளிடுதல்

அ = உள்ளீடு("ஓர் எண்ணைச் சொல்லுங்கள் ")

## இரண்டாவது எண்ணை உள்ளிடுதல்

ஆ = உள்ளீடு("இன்னொரு எண்ணைச் சொல்லுங்கள் ")

## உள்ளிட்டப்பட்ட இரு எண்களும் சமமா என்று ஒப்பிட்டுப் பார்த்தல்

@(அ == ஆ) ஆனால்
    பதிப்பி "இரு எண்களும் சமம்!"
இல்லை

## அவற்றில் ஏதேனும் ஒன்று பெரியதா என்று ஒப்பிட்டுப் பார்த்தல்

@(அ > ஆ) ஆனால்
    பதிப்பி "முதல் எண் பெரியது!"
இல்லை
    பதிப்பி "இரண்டாம் எண் பெரியது!"
முடி

முடி
```

5.2 பயிற்சி

இப்போது, உங்களுக்கு ஒரு பயிற்சி. கீழ்க்காணும் தேவைக்கேற்ப "எழில்" மொழியில் ஒரு நிரல் எழுதுங்கள், ஆங்காங்கே உரிய விளக்கங்களுடன் அது அமையவேண்டும்:

- * இன்று என்ன கிழமை என்று கேட்டுத் தெரிந்துகொள்ளவேண்டும்
- * அது சனி அல்லது ஞாயிறு எனில், விடுமுறை நாளை அனுபவிக்குமாறு வாழ்த்தவேண்டும்

விடை:

```
## இந்த நிரலை எழுதியவர்: கவிதா

## கிழமை உள்ளிடப்படுகிறது

கிழமை = உள்ளீடு("இன்று என்ன கிழமை?")

## இது சனிக்கிழமையா?

@(கிழமை == "சனி") ஆனால்
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

பதிப்பி "மகிழ்ச்சி, விடுமுறை நாளை அனுபவியுங்கள்!"

இல்லை

இது ஞாயிற்றுக்கிழமையா?

@(கிழமை == "ஞாயிறு") ஆனால்

பதிப்பி "மகிழ்ச்சி, விடுமுறை நாளை அனுபவியுங்கள்!"

இல்லை

பதிப்பி "நன்றி, இந்த நாள் இனிய நாளாகட்டும்!"

முடி

முடி

6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements)

6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements)

கணினியின் மிகப் பெரிய பலங்களில் ஒன்று, சில செயல்களைத் திரும்பத் திரும்பச் செய்வது.

ஆங்கிலத்தில் இதனை Loop என்பார்கள். தமிழில் 'வளையம்' என்று சொல்லலாம். அதாவது, ஒரு வளையத்தின்மீது ஊர்ந்து செல்லும் எழும்பைப் போல், சுற்றிச் சுற்றி வருவது.

ஆனால், அப்படி நிரந்தரமாகச் சுற்றிக்கொண்டே இருக்கமுடியாதல்லவா? ஒருகட்டத்தில் அதிலிருந்து வெளியே வரவேண்டும். அதற்கு ஒரு நிபந்தனை(Condition)யைத் தரவேண்டும்.

6.1 வரை குறிச்சொல் (While statement)

ஓர் உதாரணத்தோடு பார்ப்போம். எழில் மொழியில் 'வரை' என்ற குறிச்சொல்லைப் பயன்படுத்தி (ஆங்கிலத்தில் "While") ஓர் எளிய நிரல் எழுதுவோம்.

அ = 0

@(அ < 5) வரை

பதிப்பி "வணக்கம்!"
அ = அ + 1

முடி

இந்த நிரலை வரி வரியாகப் படித்துப் புரிந்துகொள்வோம்:

1. முதலில் "அ" என்ற மாறியில் 0 என்ற எண்ணைச் சேமிக்கிறோம்
2. அடுத்து, அந்த "அ" என்ற மாறியில் உள்ள எண் ஐந்தைவிடக் குறைவா என்று பார்க்கிறோம் (ஆம்!)
3. ஆம் எனில், நாம் இரண்டு வேலைகளைச் செய்கிறோம்:
 4. திரையில் வணக்கம் என்று அச்சிடுகிறோம்
 5. "அ" என்ற மாறியுடன் ஒன்றைக் கூட்டுகிறோம். அதாவது $அ = 0 + 1 = 1$
6. மறுபடி, "அ" என்ற மாறியில் உள்ள எண் ஐந்தைவிடக் குறைவா என்று பார்க்கிறோம் (ஆம்!)
7. மறுபடி, திரையில் "வணக்கம்" என்று அச்சிடுகிறோம், "அ" என்ற எண்ணுடன் ஒன்றைக் கூட்டுகிறோம் $அ = 1 + 1 = 2$
8. இப்படியே 6, 7 இரண்டையும் திரும்பத் திரும்ப செய்கிறோம், "அ" என்ற எண்ணில் 3, 4, 5 என்ற எண்கள் பதிவாகின்றன
9. இப்போது "அ" என்ற மாறியில் பதிவாகியிருக்கும் எண் ஐந்தைவிடக் குறைவாக இல்லை, ஆகவே நிரலை அத்துடன் நிறைவு செய்துவிடுகிறோம்

இப்போது கணக்குப் போட்டுச் சொல்லுங்கள், "வணக்கம்" என்ற சொல் எத்தனைமுறை அச்சாகியிருக்கிறது?

ஐந்து முறை, இல்லையா?

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

அதுதான் "வரை" என்ற குறிச்சொல்லின் வேலை. நாம் குறிப்பிடும் நிபந்தனை ($a < 5$) உண்மையாக உள்ளவரை, அதே வேலையைத் திரும்பத் திரும்பச் செய்துகொண்டிருக்கும்.

கிட்டத்தட்ட இதேபோல், "செய்", "முடியேனில்" என்ற இரண்டு குறிச்சொற்கள் இருக்கின்றன. ஆங்கிலத்தில் இவற்றை "Do Until Statement" என்று அழைப்பார்கள்.

இதற்கு ஓர் உதாரணம் இதோ:

அ = 0

செய்

பதிப்பி "வணக்கம்!"
அ = அ + 1

முடியேனில் $(a < 5)$

இந்த நிரலும் முந்தைய நிரலைப்போலவேதான் உள்ளது. ஆனால், இதனை எழுதி இயக்கிப் பார்த்தால் ஒரு மிகப் பெரிய வித்தியாசம் தெரியவரும், இங்கே "வணக்கம்" என்ற சொல் ஐந்து முறை அல்ல, ஆறு முறை அச்சாகும்.

ஏன் தெரியுமா?

"வரை" என்ற குறிச்சொல்லைப் பயன்படுத்தும்போது, உங்கள் கணினி " $a < 5$ " என்ற நிபந்தனையை முதலில் பரிசோதித்துவிட்டு, அதன்பிறகுதான் நாம் சொல்வதை அச்சிடும். ஆகவே, $a = 5$ என்று ஆனதும், உடனடியாக அச்சிடுவதை நிறுத்திவிடும்.

ஆனால் "செய்", "முடியேனில்" ஆகிய குறிச்சொற்கள் அப்படியல்ல, இவற்றைப் பயன்படுத்தினால் உங்கள் கணினி முதலில் அச்சிட்டுவிட்டு, அதன்பிறகுதான் " $a < 5$ " என்ற நிபந்தனையைப் பரிசோதிக்கும். ஆகவே, $a = 5$ என்று ஆனபிறகும், கூடுதலாக இன்னொருமுறை (அதாவது, 6 வது முறையாக) "வணக்கம்" என்று அச்சிட்டுவிட்டுதான் நிற்கும்.

இதனால், நீங்கள் நிரல் எழுதும்போது எப்போது "வரை" பயன்படுத்தவேண்டும், எப்போது "செய்", "முடியேனில்" கூட்டணியைப் பயன்படுத்தவேண்டும் என்று யோசித்துத் தீர்மானிக்கவேண்டும். தவறுகளைத் தவிர்க்கவேண்டும்.

இப்போது, உங்களுக்கு ஒரு [பயிற்சி](#):

- திரையில் ஓர் எண்ணைக் கேட்கவேண்டும்
- அந்த எண் ஐம்பதாகவோ, அல்லது அதைவிடக் குறைவான ஓர் எண்ணாகவோ இருந்தால், அதற்குப் பதில் இன்னொரு எண்ணைக் கேட்கவேண்டும்
- ஐம்பதைவிடப் பெரிய ஓர் எண் வரும்வரை தொடர்ந்து கேட்டுக்கொண்டே இருக்கவேண்டும்

விடை:

அ = உள்ளீடு ("ஏதாவது ஓர் எண்ணைத் தாருங்கள் ")

6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements)

@ (அ <= 50) வரை

அ = உள்ளீடு ("இந்த எண் வேண்டாம். இன்னொரு எண்ணைத் தாருங்கள் ")

முடி

பதிப்பி "நன்றி!"

அடுத்து, நாம் "தேர்ந்தெடு" என்ற குறிச்சொல்லைக் கற்றுக்கொள்ளப்போகிறோம். ஆங்கிலத்தில் இதனை **Case Statement** என்று அழைப்பார்கள்.

இதை ஓர் உதாரணத்துடன் பார்த்தால் புரியும். உங்கள் வகுப்பில் உள்ள ஒவ்வொருவரின் பெயரைக் கேட்டு, அவர் ஆணாக இருந்தால் "செல்வன்" என்றும், பெண்ணாக இருந்தால் "செல்வி" என்றும் அடைமொழியோடு அழைத்து வணக்கம் சொல்லவேண்டும். அதற்கு ஒரு நிரல் எழுதுவோம்.

6.2 தேர்ந்தெடு, நிறுத்து, தொடர் - குறிச்சொல் (Switch-Case, Break, Continue, statement)

பெயர் = உள்ளீடு ("உங்கள் பெயர் என்ன? ")

பாலினம் = உள்ளீடு ("நீங்கள் ஆணா? பெண்ணா? ")

@ (பாலினம்) தேர்ந்தெடு

தேர்வு @ ("ஆண்")

பதிப்பி "வணக்கம் செல்வன். ", பெயர்
நிறுத்து

தேர்வு @ ("பெண்")

பதிப்பி "வணக்கம் செல்வி. ", பெயர்
நிறுத்து

முடி

எளிமையான நிரல்தான். இல்லையா?

- * முதலில் உங்கள் நண்பருடைய பெயர் மற்றும் பாலினத்தைக் கேட்டுக்கொள்கிறோம்
- * பின்னர் பாலினத்தை அடிப்படையாக வைத்து, ஒவ்வொரு தேர்வாகப் பார்க்கிறோம்
- * ஆண் எனில் "வணக்கம் செல்வன். " என்று சொல்லி அவர் பெயரை அச்சிடுகிறோம், அதோடு நிரலை நிறுத்திவிடுகிறோம், இதற்கு "நிறுத்து" என்ற குறிச்சொல் பயன்படுகிறது
- * ஒருவேளை அப்படி நிறுத்தாமல் நிரலை மேலும் தொடரவேண்டும் என்றால், அதற்குத் "தொடர்" என்ற குறிச்சொல்லைப் பயன்படுத்தலாம்
- * உங்கள் நண்பர் பெண் எனில் "வணக்கம் செல்வி. " என்று சொல்லி அவர் பெயரை அச்சிடுகிறோம், அதோடு நிரலை நிறுத்திவிடுகிறோம்
- * நிறைவாக "முடி" என்ற குறிச்சொல்லைப் பயன்படுத்தி வளையத்தை நிறைவு செய்கிறோம்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

இப்படி இரண்டு, மூன்று "தேர்வு"கள்மட்டும் இருந்தால், எளிதாக நிரல் எழுதிவிடலாம், நிறைய "தேர்வு"கள் இருந்துவிட்டால், ஒவ்வொன்றாக யோசித்து எழுதுவது சிரமமாயிற்றே.

பிரச்சனையில்லை, அதற்காக "ஏதேனில்" என்ற குறிச்சொல் உள்ளது, அதற்குமேலே தரப்பட்டுள்ள எந்தத் "தேர்வு"ம் பொருந்தாவிட்டால், இங்கே உள்ள நிரல் வரிகள் நிறைவேற்றப்படும். உதாரணமாக:

ஊர் = உள்ளீடு ("உங்கள் ஊர் என்ன? ")

@(ஊர்) தேர்ந்தெடு

தேர்வு @ ("சென்னை")

பதிப்பி "சென்னையில் உள்ள மெரீனா கடற்கரை அழகானது!"
நிறுத்து

தேர்வு @ ("கோவை")

பதிப்பி "கோவையின் தமிழ் மிக இனிமையானது!"
நிறுத்து

தேர்வு @ ("மதுரை")

பதிப்பி "சிறந்த கோயில்கள் நிறைந்தது மதுரை!"
நிறுத்து

ஏதேனில்

பதிப்பி "உங்கள் ஊர்பற்றி அறிந்ததில் மகிழ்ச்சி!"
நிறுத்து

முடி

அவ்வளவுதான், எல்லா ஊர்களையும் பட்டியல் போடாமல், "ஏதேனில்" என்ற குறிச்சொல்லைப் பயன்படுத்திவிட்டோம். சென்னை, கோவை, மதுரைதவிர மற்ற அனைத்து ஊர்களுக்கும் இந்த வரிதான் அச்சிடப்படும்.

உங்கள் வீட்டுச் சமையலறையில் சிறிய, பெரிய தட்டுகள் இருந்தால், ஒரு தட்டை இன்னொன்றுக்குள் பொருத்தலாம் அல்லவா? அதுபோல, எழில் கணினி மொழியில் இந்த வளையங்களை ஒன்றுக்குள் ஒன்று என அமைக்கலாம். அதனைப் 'பின்னல் வளையம்' (Nested Loop) என்பார்கள்.

ஆனால் ஒன்று, பின்னல் வளையம் என்பது, ஒரு தட்டுக்குள் இன்னொன்றை வைப்பதுபோல், உள்ளே இருக்கும் வளையம் கச்சிதமாக வெளி வளையத்துக்குள் பொருந்தவேண்டும். இல்லாவிட்டால், உங்கள் நிரல் வேலை செய்யாது.

பின்னல் வளையங்களை எப்படி அமைப்பது என்பதற்கு உதாரணமாக இந்த நிரலைப் பாருங்கள்:

அ = 1

@(அ <= 10) வரை

6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements)

```
ஆ = 1
@ (ஆ <= 10) வரை
    இ = அ * ஆ
    ஈ = str(அ) + " * " + str(ஆ) + " = " + str(இ)
    பதிப்பி ஈ
    ஆ = ஆ + 1
முடி
அ = அ + 1
முடி
```

இந்த நிரல் $1 * 1 = 1$ என்பதில் தொடங்கி $10 * 10 = 100$ என்பதுவரையிலான பெருக்கல் வாய்ப்பாட்டை உங்களுடைய திரையில் அச்சிடும்.

இங்கே "அ" என்ற மாறியை அடிப்படையாகக் கொண்ட வளையம் வெளியே உள்ளது, "ஆ" என்ற இன்னொரு மாறியை அடிப்படையாகக் கொண்ட வளையம் உள்ளே இருக்கிறது, அது முடிந்தபிறகுதான் வெளிவளையமும் முடிகிறது என்பதைக் கவனித்துப் பாருங்கள்.

6.3 பயிற்சி

இப்போது, உங்களுக்கு ஒரு பயிற்சி:

- * நீங்கள் ஒன்றிலிருந்து நூறுக்குள் ஓர் எண்ணைத் தீர்மானித்துக்கொள்ளவேண்டும் (இதற்கு நீங்கள் "randint(1,100)" என்ற கட்டளையைப் பயன்படுத்தலாம்)
- * மற்றவர்களை அதை ஊகிக்கச் செய்யவேண்டும், அவர்களுக்கு உதவியாக, சிறு துப்பு(Clue)கள் தரலாம்
- * பத்து முயற்சிகளுக்குள் அவர்கள் அந்த எண்ணைச் சரியாக ஊகித்துவிட்டால், அவர்கள் வெற்றி பெற்றார்கள் என அறிவிக்கவேண்டும்

விடை:

```
எண் = randint(1,100)
```

```
வாய்ப்பு = 0
```

```
@ (வாய்ப்பு < 10 ) வரை
```

```
    ஊகித்தஎண் = உள்ளீடு ( "ஒன்றிலிருந்து நூறுக்குள் உள்ள ஏதோ ஓர் எண்ணை நான் மனத்தில் நினைத்துள்ளேன். அது என்ன என்று உங்களால் ஊகிக்கமுடியுமா? ")
```

```
    வாய்ப்பு = வாய்ப்பு + 1
```

```
    @ ( எண் == ஊகித்தஎண் ) ஆனால்
        பதிப்பி "வாழ்த்துக்கள்! சரியான பதில்!"
        exit(0)
முடி
```

```
@ ( எண் < ஊகித்தஎண் ) ஆனால்
    பதிப்பி "நீங்கள் சொல்லும் எண் தவறு, நான் நினைத்த எண்ணைவிட அது பெரியது!"
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

இல்லை

பதிப்பி "நீங்கள் சொல்லும் எண் தவறு, நான் நினைத்த எண்ணைவிட அது சிறியது"
முடி

பதிப்பி "கவலை வேண்டாம், இன்னும் ", (10 - வாய்ப்பு), " வாய்ப்புகள் உள்ளன. நீங்கள் நிச்சயம் வெல்லலாம், மீண்டும் முயற்சி செய்யுங்கள்!"

முடி

பதிப்பி "மன்னிக்கவும், நீங்கள் பத்து வாய்ப்புகளுக்குள் சரியான எண்ணைக் கண்டுபிடிக்கவில்லை!"

6.4 ஒவ்வொன்றாக குறிச்சொல் (For-Each statement)

அடுத்து, 'ஒவ்வொன்றாக' என்கிற ஒரு குறிச்சொல்லைப் பார்க்கவிருக்கிறோம். ஆங்கிலத்தில் இதனை 'For Each Loop' என்று அழைப்பார்கள்.

உதாரணமாக, ஒன்று முதல் ஆறு வரை உள்ள எண்களை அச்சிடவேண்டும் என்றால், அதை இப்படிச் செய்யலாம்:

எண்கள் = [1, 2, 3, 4, 5, 6]

@ (எண்கள் இல் இவ்வெண்) ஒவ்வொன்றாக
பதிப்பி இவ்வெண்

முடி

இங்கே நாம் 'எண்கள்' என்ற பட்டியலில் ஆறு எண்களை வைத்துள்ளோம். பின்னர் அவற்றை ஒவ்வொன்றாக எடுத்து 'இவ்வெண்' என்ற பெயரில் சேமிக்கிறோம், பின் அதனை அச்சிடுகிறோம்.

ஒருவேளை, ஆறுக்குப் பதில் நூறு எண்கள் இருந்தால்? அத்துனை நீளமாக எழுதுவது சிரமம் ஆயிற்றே!

உண்மைதான். அதற்காகவே 'எழில்' மொழியில் 'ஆக' என்ற குறிச்சொல் உள்ளது. இதனைப் பயன்படுத்தினால், ஒவ்வொன்றாக எழுதாமல், அதனை ஒரு கணக்காகச் செய்துவிடலாம், இப்படி:

@ (எண் = 1, எண் <= 100, எண் = எண் + 1) ஆக
பதிப்பி எண்

முடி

மிக எளிய நிரல்தான். ஒவ்வொரு வரியாகப் படித்தால் தெளிவாகப் புரிந்துவிடும்:

- * முதலில் 'எண்' என்ற பெயரில் 1 என்ற எண்ணைச் சேமிக்கிறோம்
- * பிறகு, 'எண் <= 100' என்று நிபந்தனை விதிக்கிறோம்
- * நிறைவாக, எண்ணை ஒவ்வொன்றாக அதிகரிக்கிறோம்
- * ஆக, 'எண்' என்பது 1, 2, 3, 4, 5, 6 ... என படிப்படியாக அதிகரிக்கும், திரையில் அச்சிடப்படும்
- * அந்த 'எண்' 101 ஆக மாறியதும், 'எண் <= 100' என்ற நிபந்தனை தவறாகிவிடுகிறது,

6 - மடக்குச்செயல் மற்றும் வாக்கியம் (Loops and more statements)

ஆகவே, அதற்குமேல் அச்சிடுவதை நிறுத்திவிடுகிறோம்

இப்போது, உங்களுக்கு ஒரு பயிற்சி:

* ஒன்று முதல் ஆயிரம் வரை உள்ள எண்களில் ஒற்றைப்படை எண்களைமட்டும் திரையில் அச்சிடவேண்டும்

* அடுத்தபடியாக, ஆயிரத்திலிருந்து ஒன்றுவரை உள்ள எண்களில் இரட்டைப்படை எண்களைமட்டும் திரையில் அச்சிடவேண்டும்

* இதற்கு நீங்கள் "%" என்ற குறியீட்டைப் பயன்படுத்தலாம், அது இரு எண்களை வகுத்து, அதன்பிறகு மீதமுள்ள எண்ணைமட்டும் தரும்.

விடை:

பதிப்பி "ஒற்றைப்படை எண்கள்"

@(எண்=1, எண் <=1000, எண் = எண் + 1) ஆக

மீதம் = எண் % 2

@(மீதம் == 1) ஆனால்

ஒற்றைப்படை எண்

பதிப்பி எண்

முடி

முடி

பதிப்பி "இரட்டைப்படை எண்கள்"

@(எண்=1000, எண் >= 1, எண் = எண் - 1) ஆக

மீதம் = எண் % 2

@(மீதம் == 0) ஆனால்

#இரட்டைப்படை எண்

பதிப்பி எண்

முடி

முடி

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

7 - கொஞ்சம் 'வரைந்து' பார்ப்போமா? (Drawing pictures - Turtle graphics)

இதுவரை 'எழில்' மொழியைக் கொண்டு நிறைய விஷயங்களை 'எழுதி'ப் பார்த்துவிட்டோம், ஒரு மாறுதலுக்காக, கொஞ்சம் 'வரைந்து' பார்ப்போமா?

வேடிக்கை இல்லை, நிஜமாகவே, 'எழில்' மொழியைக் கொண்டு படங்கள் வரையமுடியும். அடிப்படையான கோடு, வட்டம் போன்றவற்றில் தொடங்கி, கொஞ்சம் மெனக்கெட்டால் முழுமையான ஒவியங்களைக்கூட வரையமுடியும். அதைதான் இந்த அத்தியாயத்தில் கற்றுக்கொள்ளப்போகிறோம்!

இதற்கு நாம் பல புதிய "எழில்" குறிச்சொற்களைப் பயன்படுத்தவேண்டியிருக்கும். அவற்றில் சிலவற்றை முதலில் பட்டியல் போட்டுவிடுவோம்:

- * முன்னாடி
- * வலது
- * இடது
- * பின்னாடி
- * எழுதுகோல்மேலே
- * எழுதுகோல்கீழே

இந்தப் பெயர்களைப் பார்த்தாலே, அவை என்ன செய்யும் என்பது உங்களுக்கு ஓரளவு புரியும். அவற்றைப் பயன்படுத்தி நிரல் எழுதத் தொடங்குமுன், 'எழுதுகோல்' என்றால் என்ன?

சாதாரணமாக நாம் ஒரு காகிதத்தில் படம் வரையும்போது, எழுதுகோல் என்பது பேனா, அல்லது பென்சில். அதைக் காகிதத்தின்மீது வைத்துப் பல திசைகளில் இழுக்கிறோம். அப்படியே படம் உருவாகிறது.

கணினியிலும் அதுபோல் ஓர் எழுதுகோல் இருப்பதாகக் கற்பனை செய்துகொள்ளுங்கள், உங்களுடைய "எழில்" நிரல்மூலமாக, அந்த எழுதுகோலைப் பல திசைகளில் நகர்த்துகிறீர்கள், படம் வரைகிறீர்கள்.

உதாரணமாக, ஓர் எளிய பயிற்சி. கோடு ஒன்றை வரையப் பழகுவோம்.

காகிதத்தில் கோடு வரைவது சலபம். எழுதுகோலை ஓர் இடத்தில் வைத்து, அங்கிருந்து வலதுபக்கமாகக் கொஞ்சம் நகர்த்தினால் கோடு உருவாகிவிடும். அவ்வளவுதான்.

கணினியிலும் அதையே செய்யலாம். இதோ இப்படி:

முன்னாடி (50)

அவ்வளவுதான், திரையின் மையத்தில் உள்ள எழுதுகோல், 50 புள்ளிகள் வலதுபக்கமாக நகரும். கோடு உருவாகிவிடும்.

அதே கோட்டை மேலிருந்து கீழாக வரைவதென்றால்? இதோ இப்படி:

வலது (90)

7 -கொஞ்சம் 'வரைந்து' பார்ப்போமா? (Drawing pictures - Turtle graphics)

முன்னாடி (50)

7.1 வலது, இடது வரைதல் - (Drawing Forward, Backward, Right, Left)

இங்கே நாம் எழுதுகோலை வலதுபக்கமாக 90 பாகைகள் திருப்புகிறோம் (அதாவது கீழ்நோக்கி), பின்னர் 50 புள்ளிகள் முன்னே நகர்கிறோம் (அதாவது, கீழே), இதனால் மேலிருந்து கீழே ஒரு கோடு உருவாகிவிடுகிறது. மிகச் சுலபம்!

இதையே இன்னும் கொஞ்சம் நீட்டி, ஒரு முழுச் சதுரம் வரைவோமா?

முன்னாடி (50)

வலது (90)

முன்னாடி (50)

வலது (90)

முன்னாடி (50)

வலது (90)

முன்னாடி (50)

வலது (90)

இந்தச் சதுரம் அழகாக இருக்கிறது, பக்கத்திலேயே இன்னொரு பெரிய சதுரம் வரையவேண்டும். எப்படி?

7.2 எழுதுகோல்மேலே (Pen up, Pen down)

அதற்கு நாம் "எழுதுகோல்மேலே" மற்றும் "எழுதுகோல்கீழே" என்ற கட்டளைகளைப் பயன்படுத்தவேண்டும். இதன்மூலம் உங்கள் திரையில் எழுதுகோல் நகரும், ஆனால் படம் வரையாது, இதோ இப்படி:

எழுதுகோல்மேலே ()

முன்னாடி (200)

எழுதுகோல்கீழே ()

முன்னாடி (100)

வலது (90)

முன்னாடி (100)

வலது (90)

முன்னாடி (100)

வலது (90)

முன்னாடி (100)

இப்போது, இந்த நிரல் முழுவதையும் இயக்கிப் பார்த்தால், உங்கள் திரையில் இரண்டு சதுரங்களைக் காணலாம்.

சதுரம் வரைவதற்காக இவ்வளவு தூரம் சிரமப்படாமல், எளிதாக ஒரு "வரை" குறிச்சொல்லைப் பயன்படுத்தியும் வரையலாம், இதோ இப்படி:

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

```
அ = 0
@ (அ < 4) வரை
    முன்னாடி(50)
    வலது(90)
    அ = அ + 1
முடி
```

இந்தப் படம் அழகாக இருக்கிறது. அடுத்து, நான் இதை அழித்துவிட்டு இன்னொரு படத்தை வரைவதற்கு விரும்புகிறேன். அது எப்படி?

அதற்கு நீங்கள் `turtle_reset` என்ற கட்டளையைப் பயன்படுத்தவேண்டும். இதோ இப்படி:

```
turtle_reset()
```

அடுத்து பல வண்ணங்களில் இன்னும் பெரிய படங்களை வரைவதற்குப் பழகுவோம். இதற்கு நீங்கள் பயன்படுத்தவேண்டிய குறிச்சொல், `turtle_color`.

உதாரணமாக, ஒரு சிவப்புச் சதுரம் வரையலாமா? இதோ இப்படி:

```
அ = 0
turtle_color("red")
@ (அ < 4) வரை
    முன்னாடி(50)
    வலது(90)
    அ = அ + 1
முடி
```

7.3 நிறங்கள், வட்டங்கள் Colors, Circles

முன்பு பார்த்த அதே நிரல்தான், கூடுதலாக, `turtle_color` என்ற குறிச்சொல்லைத் தந்து, சிவப்பு நிறத்தில் வரையும்படி கணினிக்குச் சொல்கிறோம். அவ்வளவுதான் வித்தியாசம்.

அடுத்து, சிவப்பு நிறத்தில் ஒரு வட்டம் வரையலாமா? அதற்கு நாம் கூடுதலாகப் பயன்படுத்தவேண்டிய குறிச்சொற்கள் இரண்டு: `turtle_fill` & `turtle_circle`

```
turtle_color("red")
turtle_fill(True)
turtle_circle(100)
```

ஒருவேளை நீங்கள் அரைவட்டம் வரைய விரும்பினால், `turtle_circle` என்ற குறிச்சொல்லுடன், 180 பாகை என்கிற எண்ணையும் சேர்த்துக் கொடுங்கள்:

```
turtle_circle(100, 180)
```

7 -கொஞ்சம் 'வரைந்து' பார்ப்போமா? (Drawing pictures - Turtle graphics)

அப்படியானால், கால்வட்டத்துக்கு என்ன செய்வீர்கள்?

```
turtle_circle(100, 90)
```

மிக எளிமைதான். இல்லையா?

அடுத்து, கொஞ்சம் சிரமமான ஒரு வடிவத்தை வரைவோம்: பச்சை நிற நட்சத்திரம்!

இந்த நிரல் பார்ப்பதற்குச் சற்று சிரமமாக இருக்கும், ஆனால், காகிதத்தில் நட்சத்திரம் வரைந்து, உங்கள் பேனா எந்தத் திசையில் எவ்வளவு தூரம் நகர்கிறது என்று ஒருமுறை கவனித்தால் தெளிவாகப் புரிந்துவிடும்:

```
turtle_color("green")
வலது(36)
முன்னாடி(100)
வலது(144)
முன்னாடி(100)
வலது(144)
முன்னாடி(100)
வலது(144)
முன்னாடி(100)
வலது(144)
முன்னாடி(100)
```

இதுவரை நீங்கள் கற்றுக்கொண்ட விஷயங்களைப் பயன்படுத்தி, ஒரே செயலைப் பலமுறை செய்யும் கணினியின் திறமையையும் ஒருங்கிணைத்தால், ஏராளமான புதுப்புது விஷயங்களை நீங்கள் வரைந்து பார்த்துக் கலக்கமுடியும்.

இதை நிரூபிக்கும்வண்ணம், உங்களுக்கு இப்போது ஒரு பயிற்சி. "யின் யாங்" என்ற பிரபலமான சின்னத்தைப்பற்றிக் கேள்விப்பட்டிருப்பீர்கள். தெரியாது என்றால், இணையத்தில் தேடுங்கள், "எழில்" மொழியில் அதனை வரைந்து பாருங்கள்

விடை:

```
turtle_color("black")
turtle_fill(True)
turtle_circle(100, 180)
turtle_circle(200, 180)
இடது(180)
turtle_circle(-100, 180)
turtle_color("white")
turtle_fill(True)
turtle_color("black")
இடது(90)
எழுதுகோல்மேலே()
முன்னாடி(200*0.375)
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

```
வலது(90)  
எழுதுகோல்கீழே()  
turtle_circle(200*0.125)  
இடது(90)  
turtle_fill(False)  
எழுதுகோல்மேலே()  
பின்னாடி(200*0.375)  
எழுதுகோல்கீழே()  
இடது(90)
```

8 -நிரல்பாகம் (Functions – Composition)

8 - நிரல்பாகம் (Functions – Composition)

இதுவரை 'எழில்' மொழியைப் பயன்படுத்திப் பல்வேறு உபயோகமான நிரல்களை எழுதினோம், அவற்றைப் பயன்படுத்திப் புரிந்துகொண்டோம்.

அடுத்தகட்டமாக, பொதுப் பயன்பாட்டுக்குரிய சில விசேஷ நிரல்களை எழுதப் பழகுவோம். இவற்றை நீங்கள் ஒருமுறை எழுதிவிட்டால் போதும், பிறகு வேண்டியபோதெல்லாம் அழைத்துப் பயன்படுத்தலாம்.

உதாரணமாக, உங்கள் வீட்டில் ஊறுகாய் இருக்கிறது, அதை ஒருமுறை தயாரித்து வைத்துவிடுகிறீர்கள், பிறகு எப்போது அதைச் சாப்பிட விரும்பினாலும் ஜாடியைத் திறந்து எடுத்துப் போட்டுக்கொள்கிறோம், ஒவ்வொருமுறையும் ஊறுகாயைப் புதிதாகச் சமைக்கவேண்டியதில்லை.

அதே சமையலறையிலிருந்து இன்னொரு உதாரணம், ஒருவர் ரசம் சமைக்கும்போது, ஒரு குறிப்பிட்ட செய்முறையைப் பின்பற்றுவார். அதன் நடுவே 'ரசப்பொடியைப் போடவும்' என்று இருக்கும்.

ரசப்பொடி என்பதை, அவர் இப்போது தயாரிப்பதில்லை, ஏற்கெனவே எப்போதோ தயாரித்துவைத்துவிட்டார், பின்னர் தேவைப்படும்போது சட்டென்று எடுத்துப் பயன்படுத்துகிறார், அதன்பிறகு, செய்முறையில் இருக்கும் மற்ற விஷயங்களைத் தொடர்ந்து செய்கிறார்.

அதுபோல, ஒரு நிரலின் நடுவே, சில விஷயங்கள் நாம் அடிக்கடி செய்கிறவையாக இருக்கும், அவற்றை ஒவ்வொருமுறையும் தனித்தனியே எழுதிக்கொண்டிருக்காமல், ஒருமுறை எழுதிவைத்துவிட்டு, தேவைப்படும்போது அழைத்துப் பயன்படுத்தலாம், பிறகு, நம்முடைய நிரலைத் தொடர்ந்து எழுதலாம்.

உதாரணமாக, இரண்டு எண்களில் எது பெரியது என்று கண்டுபிடிப்பதற்கான எளிய நிரல் ஒன்றை எழுதுவோம்:

```
எண்1 = உள்ளீடு("ஓர் எண்ணைச் சொல்லுங்கள்: ")
எண்2 = உள்ளீடு("இன்னொரு எண்ணைச் சொல்லுங்கள்: ")

@(எண்1 > எண்2) ஆனால்
    பதிப்பி "நீங்கள் தந்தவற்றுள் பெரிய எண்: ", எண்1
இல்லை
    பதிப்பி "நீங்கள் தந்தவற்றுள் பெரிய எண்: ", எண்2
முடி
```

மிக எளிய நிரல் இது. கணினிப் பயனாளரிடமிருந்து இரு எண்களைக் கேட்டு வாங்கி, அதில் எது பெரியதோ அதைமட்டும் திரையில் அச்சிடுகிறது.

இந்தப் 'பெரிய எண்ணைக் கண்டுபிடித்தல்' என்ற செயல், நம்முடைய பல கணக்குகளில் அடிக்கடி வரும், அப்போதெல்லாம் இந்த நிரலை மீண்டும் மீண்டும் எழுதிக்கொண்டிருக்காமல், ஒருமுறை எழுதிவிட்டுப் பின்னர் அதைப் பலமுறை பயன்படுத்தமுடியுமா என்று பார்ப்போம்.

இதற்கு நாம் 'நிரல்பாகம்' என்ற சிறப்புக் குறிச்சொல்லைப் பயன்படுத்தவேண்டும். ஆங்கிலத்தில் இதனை 'Function' என்று அழைப்பார்கள்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

நிரல்பாகம் பெரியது (எண்1, எண்2)

```
@ (எண்1 > எண்2) ஆனால்  
    பின்கொடு எண்1  
இல்லை  
    பின்கொடு எண்2  
முடி
```

முடி

இங்கே நாம் என்ன செய்திருக்கிறோம்? படிப்படியாக அலசுவோம்:

1. "பெரியது" என்ற பெயரில் ஒரு நிரல்பாகத்தைத் தொடங்கியிருக்கிறோம்
2. இந்த நிரல்பாகத்தை அழைக்க விரும்புவோர் அதற்கு இரண்டு எண்களைத் தரவேண்டும், அவற்றை அடைப்புக் குறியினுள் எழுதியிருக்கிறோம்
3. இவற்றுள் எண்1 பெரியது என்றால், நிரல்பாகத்தை அழைத்தோருக்கு அதையே விடையாகக் கொடுக்கிறோம், இல்லாவிட்டால் எண்2 வை விடையாகத் தருகிறோம், இதற்குப் 'பின்கொடு' என்ற குறிச்சொல் பயன்படுகிறது. ஆங்கிலத்தில் இதனை 'Return' என்பார்கள்.

இங்கே நாம் நிரல்பாகத்தைமட்டும்தான் எழுதியிருக்கிறோம். அதனை இன்னும் அழைக்கவில்லை, அதாவது, இந்த நிரலை நாம் இன்னும் பயன்படுத்தவில்லை. ரசப்பொடியை எடுத்துப் போட்டுச் சமைத்தால்தானே ரசம் தயாராகும்?

அதற்கான நிரல் வரி மிக எளிது:

```
பதிப்பி பெரியது(10, 15)  
பதிப்பி பெரியது(12, 8)
```

அவ்வளவுதான், நமக்கு வேண்டிய இடங்களில் "பெரியது" என்ற சொல்லை எழுதி, அதற்கு வேண்டிய இரண்டு எண்களைக் கொடுத்தவுடன் விடை பளிச்சென்று திரும்பக் கிடைத்துவிடும். பிரமாதம், இல்லையா?

ஆனால், நிரல்பாகத்தின் உண்மையான பலன் இதுவல்ல, கொஞ்சம் சிக்கலான கணக்குகளில் அதைப் பயன்படுத்திப் பார்த்தால்தான் இதன் முழு பலமும் நமக்குத் தெரியவரும்.

8.1 தொடர்பெருக்கு (Factorial)

முதலில், Factorial எனப்படும் தொடர்பெருக்கு எண்ணைக் கணக்கிடுவதற்கு ஒரு நிரல் எழுதுவோம்.

அதற்குமுன்னால், ஓர் எண்ணின் Factorialலை எப்படிக் கண்டுபிடிப்பது? உங்கள் கணக்குப் பாடத்தைச் சற்றே நினைவுபடுத்திப் பாருங்கள்:

1. உதாரணமாக, எண் 7 எடுத்துக்கொள்வோம், இதன் Factorial 7! என்று குறிக்கப்படும்
2. 7! = அதில் தொடங்கி ஒன்று வரையிலான அனைத்து எண்களின் பெருக்குத் தொகை, அதாவது, 7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040

8 -நிரல்பாகம் (Functions – Composition)

அவ்வளவுதான். எளிமையான கணக்கு!

ஆனால், உங்களிடம் யாராவது '100! எவ்வளவு?' என்று கேட்டுவிட்டால் என்ன செய்வீர்கள்? $100*99*98...$ என்று கணக்குப் போடுவதற்கு ரொம்ப நேரமாகுமே!

கவலை வேண்டாம், எழில் மொழியும், அதன் நிரல்பாகமும் உங்கள் துணைக்கு வரும். இதோ இப்படி:

```
நிரல்பாகம் தொடர்பெருக்கு(அ)
  @ (அ == 0) ஆனால்
    பின்கொடு 1
  இல்லை
    பின்கொடு அ*தொடர்பெருக்கு(அ-1)
முடி
முடி

பதிப்பி தொடர்பெருக்கு(7)
பதிப்பி தொடர்பெருக்கு(100)
```

8.1.1. இந்த நிரல் எப்படி இயங்குகிறது?

முதலில் தொடர்பெருக்கு(7) எனத் தொடங்குகிறோம், அங்கே அது $7 * \text{தொடர்பெருக்கு}(6)$ என மாறுகிறது, பின் $7 * 6 * \text{தொடர்பெருக்கு}(5)$ என மாறுகிறது... இப்படியே தொடர்ந்து $7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040$ என்று கண்டறிந்துவிடுகிறோம்!

இதேபோல், தொடர்பெருக்கு(100)ஐயும் அழைக்கலாம், ஆனால் அதன் விடை மிக மிகப் பெரியது, இந்தப் புத்தகத்தில் எழுதுவது சாத்தியமில்லை!

8.2 மீண்டும் அழைக்கப்படும் நிரல்பாகம் (Recursion)

அடுத்து, இதேபோல் இரு எண்களுக்கு இடையே GCD அல்லது மீபொவ எனப்படுகிற மீப்பெரு பொது வகுத்தி கண்டுபிடிக்க ஒரு நிரல் எழுதுவோம்.

மறுபடியும், உங்கள் கணக்குப் பாடத்தைச் சற்றே நினைவுபடுத்திப் பாருங்கள், GCD என்றால் என்ன?

1. உதாரணமாக, எண்கள் 54, 42 ஆகியவற்றை எடுத்துக்கொள்வோம்
2. இதில் 54 என்பது 1, 2, 3, 6, 9, 18, 27 மற்றும் 54 ஆகியவற்றால் வகுபடும்
3. ஆனால் 42 என்பது 1, 2, 3, 4, 6, 8, 12, 24 ஆகியவற்றால் வகுபடும்
4. இந்த இரண்டு பட்டியலுக்கும் பொதுவான எண்கள் 1, 2, 3, 6 ஆகியவை
5. இவற்றில் பெரியது 6
6. ஆகவே, 54, 42 ன் GCD 6

எளிமையான கணக்குதான். ஆனால் இதைக் கையால் செய்வது சிரமம். திரும்பத் திரும்பப் பலமுறை வகுத்துப் பார்த்துக்கொண்டே இருக்கவேண்டும்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

அதற்குதான் நிரல்பாகம் பயன்படுகிறது, இதோ இப்படி:

```
நிரல்பாகம் மீபொம்(அ, ஆ)
  பெரிது = max(அ, ஆ)
  சிறிது = min(அ, ஆ)
  if(சிறிது == 0) ஆனால்
    பின்கொடு பெரிது
  இல்லை
    பின்கொடு மீபொம்(பெரிது - சிறிது, சிறிது)
  முடி
```

முடி

பதிப்பி "54, 42 இடையிலான மீபொம் = ", மீபொம்(54, 42)

இந்த நிரல் எப்படி இயங்குகிறது என நாம் படிப்படியாகப் பார்க்கவேண்டும்:

முதன்முறை:

மீபொம்(54, 42)
பெரிது = அவற்றில் பெரிய எண் = 54
சிறிது = அவற்றில் சிறிய எண் = 42
சிறிது == 0 இல்லை
ஆகவே, அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், இப்படி: மீபொம்(பெரிது - சிறிது, சிறிது), அதாவது, மீபொம்(54-42, 42), மீபொம்(12, 42)

இப்போது,
பெரிது = 42
சிறிது = 12
சிறிது == 0 இல்லை
அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், மீபொம்(42-12, 12), அதாவது
மீபொம்(30, 12)

இப்போது,
பெரிது = 30
சிறிது = 12
சிறிது == 0 இல்லை
அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், மீபொம்(30-12, 12) அதாவது,
மீபொம்(18, 12)

இப்போது,
பெரிது = 18
சிறிது = 12
சிறிது == 0 இல்லை
அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், மீபொம்(18-12, 12) அதாவது,
மீபொம்(6, 12)

இப்போது,
பெரிது = 6
சிறிது = 12

8 -நிரல்பாகம் (Functions – Composition)

சிறிது == 0 இல்லை
அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், மீபொம்(12-6, 6) அதாவது
மீபொம்(6, 6)

இப்போது,
பெரிது = 6
சிறிது = 6
சிறிது == 0 இல்லை
அதே மீபொம் நிரல்பாகம் மீண்டும் அழைக்கப்படும், மீபொம்(6-6, 0), அதாவது
மீபொம்(6, 0)

இப்போது,
பெரிது = 6
சிறிது = 0
சிறிது == 0
ஆகவே, GCD விடை = பெரிது = 6

அவ்வளவுதான். நாம் கையால் போட்ட அதே கணக்கைக் கணினியால் போட்டுவிட்டோம்.
நிரல்பாகத்தின் உண்மையான பலன் இப்போது தெரிந்திருக்கும். ஒருமுறை இருமுறை அல்ல, நூறு
முறை, ஆயிரம் முறைகூட இதனை நீங்கள் அழைத்துப் பயன்படுத்தலாம்!

8.3 பயிற்சி - பகா எண்கள் (Prime Numbers)

இப்போது, உங்களுக்கு ஒரு பயிற்சி:

1. ஒன்று முதல் ஐம்பது வரை உள்ள எண்களில் எவையெல்லாம் பகா எண்கள் (Prime Numbers) என்று கண்டறிந்து, திரையில் அச்சிடுங்கள்
2. பின்னர் அந்த எண்களின் கூட்டுத்தொகையையும் அச்சிடவேண்டும்
3. இந்தக் கூட்டுத்தொகை பகு எண்ணா, பகா எண்ணா என்று கண்டறியவேண்டும்
4. உங்களுக்கு உதவியாக ஒரு குறிப்பு, ஓர் எண் பகா எண்ணா என்று கண்டறிய நீங்கள் floor, fmod என்ற கணிதச் செயல்பாடுகளைப் பயன்படுத்தவேண்டியிருக்கும், இவைபற்றித் தேடிப் படித்துத் தெரிந்துகொள்ளுங்கள், அல்லது, உங்களுடைய புதிய வழிமுறையைக் கண்டறியுங்கள்

விடை:

நிரல்பாகம் பகாஎண்ணா(அ)

```
ஆ = 1
இ = floor(அ/2)
@(இ >= 2) வரை
    @(fmod(அ, இ) == 0) ஆனால்
        ஆ = 0
        நிறுத்து
    முடி
    இ = இ + 1
முடி
பின்கொடு ஆ
முடி
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

எல்லை = 50
எண் = 1
கூட்டுத்தொகை = 0

if (எண் <= எல்லை) வரை

விடை = பகாஎண்ணா(எண்)
if (விடை) ஆனால்
பதிப்பி எண்
கூட்டுத்தொகை = கூட்டுத்தொகை + எண்
முடி
எண் = எண் + 1

முடி

பதிப்பி "இதுவரை நாம் பார்த்த பகா எண்களின் கூட்டுத்தொகை: ", கூட்டுத்தொகை

கூட்டுத்தொகைவிடை = பகாஎண்ணா(கூட்டுத்தொகை)

if (கூட்டுத்தொகைவிடை) ஆனால்
பதிப்பி "இந்தக் கூட்டுத்தொகையும் ஒரு பகா எண்தான்"
இல்லை
பதிப்பி "இந்தக் கூட்டுத்தொகை பகா எண் அல்ல"

முடி

நிரல்பாகம் கணக்கில்மட்டும்தான் உதவுமா? மற்ற இடங்களில் அதனை உபயோகப்படுத்தமுடியாதா?

நன்றாக உபயோகப்படுத்தலாம். உதாரணமாக, படம் வரைவதற்கு!

ஏற்கெனவே 'எழில்' பயன்படுத்திப் பல எளிய படங்களை வரைந்திருக்கிறோம், இப்போது, கொஞ்சம் சிக்கலான ஒரு படத்தை வரைவோம், நிரல்பாகம் உதவியுடன்!

இங்கே நாம் வரையப்போவது, ஒரு செடியின் படம் (ஆங்கிலத்தில் Fern என்பார்கள்), இந்தச் செடியில் உள்ள வெவ்வேறு அளவிலான இலைகளையெல்லாம் ஒவ்வொன்றாக வரைந்துகொண்டிருக்காமல், நிரல்பாகத்தின்மூலம் சட்டென்று வரையப்போகிறோம். இதோ, இப்படி:

நிரல்பாகம் சாளரத்தில் (அ)
பதிப்பி "Window = ", அ

முடி

நிரல்பாகம் PD()
பதிப்பி "Pen Down"

முடி

நிரல்பாகம் PU()
பதிப்பி "Pen Up"

முடி

8 -நிரல்பாகம் (Functions – Composition)

நிரல்பாகம் FD(அ)

பதிப்பி "Forward ",அ

முடி

நிரல்பாகம் RT(அ)

பதிப்பி "Right ", x

முடி

நிரல்பாகம் LT(அ)

பதிப்பி "Left ",x

முடி

நிரல்பாகம் BK(அ)

பதிப்பி "Back ",அ

முடி

நிரல்பாகம் வேம்பு (அளவு, பதிவு)

@(அளவு < 1) ஆனால்

பின்கொடு 0

முடி

FD(அளவு)

RT (70)

வேம்பு (அளவு * 0.5 , -1 * பதிவு)

LT(70*பதிவு)

FD(அளவு)

LT (70)

வேம்பு (அளவு * 0.5, 1 *பதிவு)

RT(70*பதிவு)

RT (7*பதிவு)

வேம்பு (அளவு - 1, பதிவு)

LT(7*பதிவு)

BK(அளவு * 2)

முடி

சாளரத்தில் ("clearscreen")

PU()

PD()

சா = 5

BK (150*சா)

வேம்பு (25 , 1)

நாம் இதுவரை எழுதியதிலேயே மிகப் பெரிய நிரல் இதுதான். மேலோட்டமாகப் பார்ப்பதற்குக் கொஞ்சம் சிக்கலாக இருக்கும். ஆனால் உண்மையில், வேம்பு() என்ற நிரல்பாகத்தைக் கொண்டு நாம் எழுதுகோலை முன்னும் பின்னும் வலமும் இடமும் நகர்த்துகிறோம், ஓர் இலையை வரைகிறோம், பின் அடுத்த இலையை வரைகிறோம், இந்த விஷயத்தை மனத்தில் கொண்டு

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

வாசித்தால், நிரல் தெளிவாகப் புரியும்.

இந்த அத்தியாயத்தில் நாம் பார்த்தவை சில எளிய உதாரணங்கள்மட்டுமே, உண்மையில் நிரல்பாகத்தின் பயன் மிக மிகப் பெரியது, அதன்மூலம் அற்புதமான பல கணக்குகளை நொடியில் செய்துமுடிக்கலாம், வகைவகையான படங்களை வரையலாம், உங்கள் கற்பனைமட்டுமே எல்லை!

9 -கோப்பு (File I/O)

9 - கோப்பு (File I/O)

இந்தச் சிறு கையேட்டில், "எழில்" நிரல் மொழியைக் கொண்டு ஏராளமான விஷயங்களைச் செய்யத் தெரிந்துகொண்டோம். அடுத்து என்ன?

Files எனப்படும் கோப்புகளை நிரல்வழியே கையாள்வதன்மூலம் நாம் மேலும் பல பயனுள்ள நிரல்களை எழுதமுடியும். அவை வெறுமனே கற்கும் நோக்கத்துக்காகமட்டுமில்லாமல், பலருக்கும் உபயோகப்படும்.

உதாரணமாக, கண்கு-வழக்கு சார்ந்த மென்பொருள் தனிசரி நிலவரத்தை கோப்புகளால் சேமிக்கின்றன. "எழில்" மூலமாக, உங்களுடைய கணினியில் ஒரு சிறு கோப்பை உருவாக்கி, அதில் சில விஷயங்களை எழுதிப் பார்ப்போம்.

9.1 புதிய கோப்பு

முதலில் புதிய கோப்பு ஒன்றை உருவாக்குவதற்கு, நாம் "கோப்பை_திற" என்கிற சொல்லைப் பயன்படுத்தவேண்டும். இதோ இப்படி:

```
புதுக்கோப்பு = கோப்பை_திற("weekdays.txt", "w")
```

இங்கே **weekdays.txt** என்பது கோப்பின் பெயர், "**w**" என்பது நாம் அதில் ஏதோ சில விஷயங்களை எழுத(**write**)ப்போகிறோம் என்பதைக் குறிக்கும் சொல்.

இப்படி நாம் உருவாக்கிய புதிய கோப்பை, "புதுக்கோப்பு" என்ற பெயரில் சேமித்துவைக்கிறோம். பின்னர் இந்தப் பெயரைப் பயன்படுத்திப் பல விஷயங்களைச் செய்யலாம். உதாரணமாக:

```
அ = "திங்கள்"  
ஆ = "செவ்வாய்"  
இ = "புதன்"  
ஈ = "வியாழன்"  
உ = "வெள்ளி"
```

```
கோப்பை_எழுது(புதுக்கோப்பு, அ)  
கோப்பை_எழுது(புதுக்கோப்பு, ஆ)  
கோப்பை_எழுது(புதுக்கோப்பு, இ)  
கோப்பை_எழுது(புதுக்கோப்பு, ஈ)  
கோப்பை_எழுது(புதுக்கோப்பு, உ)
```

9.1.1. கோப்பை மூட

நிறைவாக, அந்தக் கோப்பை மூடவேண்டும். அதற்குப் பயன்படுத்தவேண்டிய நிரல் வரி:

```
கோப்பை_மூடு(புதுக்கோப்பு)
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

அடுத்தபடியாக, நாம் எழுதிய இந்தக் கோப்பில் என்ன இருக்கிறது என்று படிக்க விரும்புகிறோம். அதற்கு இந்தக் கட்டளைகளைப் பயன்படுத்தலாம்:

```
பழையகோப்பு = கோப்பை_திற("weekdays.txt")
வரிகள் = கோப்பை_படி(பழையகோப்பு)
பதிப்பி வரிகள்
கோப்பை_மூடு(பழையகோப்பு)
```

9.2 கோப்பை_திற படிக்க

கவனியுங்கள், இந்தமுறை கோப்பை_திற என்ற கட்டளையில் "w" என்று குறிப்பிடவில்லை, காரணம், நாம் இந்தக் கோப்பில் எதுவும் புதிதாக எழுதப்போவதில்லை, வெறுமனே படிக்கப்போகிறோம், அவ்வளவுதான்.

இப்போது உங்களுக்கு ஒரு பயிற்சி, இதுவரை நீங்கள் கற்றுக்கொண்ட விஷயங்களை வைத்து கீழே தரப்பட்டுள்ள தேவைகளுக்கேற்ப ஒரு நிரல் எழுதுங்கள்:

1. பூஜ்ஜியம் முதல் ஐநூறு வரை உள்ள Fibonacci எண்களைக் கண்டுபிடிக்கவேண்டும்
2. அவை ஒவ்வொன்றின் வர்க்கங்களை(Square)மட்டும் ஒரு கோப்பில் எழுதவேண்டும்
3. பின்னர் அவற்றைத் திரையில் அச்சிடவேண்டும்

விடை:

```
எண்1 = 0
எண்2 = 1
எல்லை = 500

புதுக்கோப்பு = கோப்பை_திற("fibonacci.txt", "w")
கோப்பை_எழுது(புதுக்கோப்பு, எண்1 * எண்1)
கோப்பை_எழுது(புதுக்கோப்பு, எண்2 * எண்2)

எண்3 = எண்1 + எண்2

@ (எண்3 <= எல்லை) வரை
    கோப்பை_எழுது(புதுக்கோப்பு, எண்3 * எண்3)

    எண்1 = எண்2
    எண்2 = எண்3
    எண்3 = எண்1 + எண்2

முடி
கோப்பை_மூடு(புதுக்கோப்பு)

# அச்சிடுதல்
```

9 -கோப்பு (File I/O)

பழையகோப்பு = கோப்பை_திற("fibonacci.txt")

வரிகள் = கோப்பை_படி(பழையகோப்பு)

பதிப்பி வரிகள்

கோப்பை_மூடு(பழையகோப்பு)

"எழில்" மொழியில் அருமையான பல பயன்பாடுகள் உண்டு என்பதை விளக்கமாகப் பார்த்துள்ளோம், இதில் நீங்கள் அடிப்படை நிரல்களை எழுதப் பழகியபிறகு, அடுத்தகட்டமாக பைதான்(Python)போன்ற "எழில்"க்கு இணையான, அதேபோன்ற வடிவமைப்பைக் கொண்ட இன்னொரு விரிவான மொழியைக் கற்கலாம்.

அப்போது, நீங்கள் ஒரு முக்கியமான விஷயத்தை உணர்வீர்கள், "எழில்" மற்றும் "பைதான்" இடையே எழுதும் விதம், மொழியின் இலக்கண சொற்றொடரியல் (Syntax) மாறுகிறதேதவிர, இவ்விரு மொழிகளிலும் நிரல் எழுதுவதற்கான அடிப்படைச் சிந்தனை (Logical Thought Process) ஒன்றுதான்.

இந்த இரு மொழிகளில்மட்டுமல்ல, உலகில் உள்ள அனைத்துக் கணினி நிரல் மொழிகளிலும், எழுதும் விதம்தான் மாறுபடும். சிந்திக்கத் தெரிந்தவர்கள் அதைக் கற்றுக்கொண்டு நிபுணராவது மிக எளிது. பின்னர் புதுப்புது மொழிகள் அறிமுகமானாலும் நாம் சுலபமாக அவற்றைத் தெரிந்துகொண்டு வெற்றி பெறலாம்.

ஆகவே, எந்த மொழியைக் கற்கிறோம் என்பதுபற்றி அதிகம் கவலைப்படவேண்டாம், நிரல் எழுத்தாளராகச் சிந்திக்கக் கற்றுக்கொண்டால் போதும், அதுவே உங்களைப் பல திசைகளுக்குக் கொண்டுசெல்லும்!

9.3 முடிவு

"எழில்" மொழியில் தொடங்கிய உங்கள் பயணம், எல்லாப் பக்கங்களிலும் விரியட்டும், உங்களுக்கு வாழ்த்துகள்! நீங்கள் ஒரு மென்பொருள் உருவாக்குபவர் ஆக முடியும்!

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

10 - எழில் மொழி உதாரணங்கள் (Examples)

உங்களுக்கு உதவும் எழில் நிரலாக்க மொழி உதாரணங்கள்

10.1 உதாரணங்கள் பதிவிறக்கம்

எடுத்துக்காட்டாக நிரல்கள் மேலும் வலைத்தளத்தில் பதிவிறக்கம் <https://github.com/arcturusannamalai/Ezhil-Lang/archive/master.zip> செய்ய. அடுத்த [ezhil_tests/](#) கோப்பில் குறிப்பிடப்படுகிறது உதாரணங்கள் பார்க்க.

வலைத்தளத்தில் இருந்து குறிப்பிடப்படுகிறது உதாரணங்கள் இந்த வலைத்தளத்தில் (website) <http://ezhillang.org/koodam/play/eval> மூலம் நீங்களே வலை உலாவியில் (browser) பயிலலாம்.

10.2 அறிமுகப்படுத்துதல்

"எழில்" மொழியைக் கற்பதில் உங்களுக்கு உதவுவதற்காகவே, நாங்கள் பல உதாரண நிரல்களை உருவாக்கியிருக்கிறோம். இவற்றை முறைப்படி பார்வையிட்டாலே போதும், நீங்களே சொந்தமாக "எழில்" நிரல்கள் எழுதத் தொடங்கிவிடலாம்.

ஆனால், எந்த வரிசையில் பார்ப்பது? எங்கே பார்ப்பது?

அதற்கு வழிகாட்டதான் இந்தக் கட்டுரை. "எழில்" மொழியின் அடிப்படைக் கூறுகளில் தொடங்கி, ஒவ்வொரு அம்சத்தையும் உதாரண நிரல்களின் வழியே கற்பது எப்படி என்று இங்கே பார்ப்போம்.

இந்தக் கட்டுரையில் தரப்பட்டுள்ள வரிகள்மட்டும்தான் என்றில்லை, இதுபோல் இன்னும் பல வரிகள் உதாரண நிரல்களில் உண்டு. அவற்றைத் தேடிப் படிக்கப் படிக்கவும், புரிந்துகொள்ளவும் உங்களுக்கு "எழில்"பற்றி நன்கு விளங்கும். புதிய நிரல்கள் எழுதுவதற்கான நம்பிக்கை, வழிமுறைகள் கிடைக்கும்.

10.3 அச்சிடுதல்

இதில் இரண்டு வகை உண்டு: எழுத்துச் சரங்களை அச்சிடுதல், நிரலில் உள்ள விவரங்களை அச்சிடுதல்.

10 -எழில் மொழி உதாரணங்கள் (Examples)

10.3.1. எழுத்துச் சரங்களை அச்சிடுதல்:

உதாரண நிரல்: [quiz.n](#)

இதில் முதலாவதாக உள்ள வரியைக் கவனியுங்கள்:

பதிப்பி "இந்தியாவின் தலைநகரம் எது?"

இந்த வாசகம், "இந்தியாவின் தலைநகரம் எது?" என்கிற எழுத்துச் சரத்தைத் திரையில் அச்சிடுகிறது. இதுபோல் நீங்கள் விரும்பும் எந்த எழுத்துச் சரத்தையும் திரையில் அச்சிடலாம்.

10.4 நிரலில் உள்ள விவரங்களை அச்சிடுதல்

உதாரண நிரல்: [string_demo.n](#)

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

பதிப்பி "முதல் சொல்லின் நீளம்", நீளம்(அ)

நாம் ஏற்கெனவே பார்த்த "பதிப்பி" வரிக்கும் இதற்கும் உள்ள வித்தியாசம் புரிகிறதல்லவா? இங்கே வழக்கமான எழுத்துச் சரத்துடன், அந்த நிரலில் உள்ள "அ" என்ற மாறியின் நீளத்தைக் கணக்கிட்டு அதையும் சேர்த்துத் திரையில் அச்சிடுகிறோம். உதாரணமாக: "முதல் சொல்லின் நீளம்10" என்பதுபோல்.

இதேமாதிரி நாம் எத்தனை விவரங்களை வேண்டுமானாலும் " பதிப்பி" வாக்கியத்தில் சேர்க்கலாம்.

உதாரண நிரல்: [calendar_days.n](#)

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

பதிப்பி month+1,"]",எடு(months,month), " has ",எடு(days,month)," days"

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

இங்கே ஒரே வரியில் பல விவரங்கள் சேர்த்து அச்சிடப்பட்டுள்ளதைப் பாருங்கள். இதுபோல் நீங்கள் உங்கள் நிரல்களின் விடைகளை வாசகர்களுக்குச் சொல்ல “பதிப்பி”யைப் பயன்படுத்தலாம்.

10.5 உள்ளிடுதல்

சில நேரங்களில், “எழில்” நிரல் இயங்குவதற்குப் பயனாளர் (User) ஒன்று அல்லது அதற்கு மேற்பட்ட எண்களையோ எழுத்துகளையோ உள்ளிடவேண்டும் (Input). உதாரணமாக, கூட்டல் நிரல் இயங்க நாம் இரண்டு எண்களைத் தரவேண்டுமல்லவா?

இதற்கு நாம் “உள்ளீடு” என்ற கட்டளையைப் பயன்படுத்துகிறோம்.

உதாரண நிரல்: leapyear.n

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

வருடம் = int(உள்ளீடு("வருடத்தின் எண்ணை உள்ளீடு செய்க (உ.தா) 1984;"))

இங்கே “உள்ளீடு” என்ற கட்டளை வந்துள்ளதால், “எழில்” நீங்கள் தந்துள்ள சரத்தைத் திரையில் அச்சிட்டுவிட்டுக் காத்து நிற்கும், பதிலுக்குப் பயனாளர் ஓர் எண்ணை உள்ளிடவேண்டும், அந்த எண் “வருடம்” என்ற மாறியில் சேமிக்கப்படும். பின்னர், அதை வைத்து அடுத்தடுத்த கணக்குகள் நிகழ்வதை மேற்சொன்ன உதாரண நிரலில் சென்று விரிவாகக் காணுங்கள்.

சுருக்கமாகச் சொன்னால், “உள்ளீடு” என்பது விவரங்களைப் பயனாளரிடமிருந்து பெறுதல், “பதிப்பி” என்பது விவரங்களைப் பயனாளருக்கு வழங்குதல். இந்த இரண்டையும் சேர்த்துப் பயன்படுத்தினால் நீங்கள் பயனாளருடன் கலந்து உரையாடும்விதமான (Interactive) நிரல்களை எழுதமுடியும்.

10.6 ஒப்பிடுதல்

இரண்டு எண்கள் அல்லது எழுத்துகளை “எழில்” நிரல்களில் எப்படி ஒப்பிடுவது?

10 -எழில் மொழி உதாரணங்கள் (Examples)

உதாரண நிரல்: [ackermann.n](#)

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

@(முதலெண் == 0) ஆனால்

இந்த வாசகம், “முதலெண்” என்ற மாறியின் மதிப்பு பூஜ்ஜியமா என்பதைப் பார்க்கிறது, அதன் அடிப்படையில் வேறு சில நிரல் வாசகங்களை நிறைவேற்றுகிறது. இதற்கு நாம் “==” என்ற குறியீட்டைப் பயன்படுத்துகிறோம்.

உதாரண நிரல்: [histogram.n](#)

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

@(முந்தினசொல்!= "") ஆனால்

இந்த வாசகத்தை முந்தைய வாசகத்துடன் ஒப்பிட்டுப் பாருங்கள். “முந்தினசொல்” என்ற மாறியில் உள்ள மதிப்பு “” ஆக இல்லையா என்பதை இந்த வாசகம் பார்க்கிறது. அதன் அடிப்படையில் வேறு சில நிரல் வாசகங்களை நிறைவேற்றுகிறது. இதற்கு நாம் “!=” என்ற குறியீட்டைப் பயன்படுத்துகிறோம்.

10.7 மேலும் சில ஒப்பீடுகள்

==, != ஆகியவைதவிர, இன்னும் பல ஒப்பீட்டுக் குறியீடுகளும் “எழில்” மொழியில் உண்டு. அவை:

- சிறியது “<”
- பெரியது “>”
- சிறியது அல்லது சமம் “<=”
- பெரியது அல்லது சமம் “>=”

உதாரண நிரல்: [rich_poor.n](#)

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

@(வருமானம் > 10000) ஆனால்

இந்த வாசகம் “ வருமானம் ” என்ற மாறியில் உள்ள எண் பத்தாயிரத்தைவிட அதிகமா என்று பார்க்கிறது. அதன் அடிப்படையில் வேறு சில நிரல் வாசகங்களை நிறைவேற்றுகிறது. இதேபோல் நாம் “<”, “<=”, “>=” ஆகிய குறியீடுகளையும் பயன்படுத்தலாம். அவற்றுக்கான பல உதாரணங்கள் இங்கே உள்ளன.

10.8 ஒன்றுக்கு மேற்பட்ட ஒப்பீடுகள்

இதுவரை நாம் பார்த்த உதாரணங்கள் அனைத்திலும், ஒரு வாசகத்தில் ஒரே ஒரு விஷயத்தைமட்டும்தான் (சமம், சமமல்ல, சிறியது, பெரியது) ஒப்பிட்டுள்ளோம். ஒருவேளை நாம் ஒன்றுக்கு மேற்பட்ட விஷயங்களை ஒப்பிடவேண்டியிருந்தால் என்ன செய்வது?

இதற்கு “எழில்” வழங்கும் குறியீடுகள் இவை:

- && (AND / மற்றும்)
- || (OR / அல்லது)

இவற்றில் “&&” குறியீட்டைப் பயன்படுத்தினால், அதற்கு முன்னால், பின்னால் இருக்கும் இரண்டு ஒப்பீடுகளும் உண்மையாகவேண்டும். “||” குறியீட்டைப் பயன்படுத்தினால், இவற்றில் ஏதேனும் ஒன்று உண்மையானாலே போதுமானது.

உதாரண நிரல்: ackermann.n

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

@((முதலெண் > 0) && (இரண்டாமெண் == 00)) ஆனால்

இந்த வாசகத்தில் இரண்டு ஒப்பீடுகள் உள்ளன: முதலெண் பூஜ்ஜியத்தைவிடப்

10 -எழில் மொழி உதாரணங்கள் (Examples)

பெரியது, இரண்டாமெண் பூஜ்ஜியத்துக்குச் சமம். இந்த இரண்டுக்கும் இடையே “&&” உள்ளது.

இதன் பொருள், முதலெண் பூஜ்ஜியத்தைவிடப் பெரியதாக இருந்து, இரண்டாமெண் பூஜ்ஜியத்துக்குச் சமமாக இருந்தால்தான் கீழே தரப்பட்டுள்ள கட்டளைகள் நிறைவேறும்.

உதாரண நிரல்: interest.n

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

```
@(மூலதனம் == 0 || வட்டி_வீதம் == 0 || வருடம் == 0 ) ஆனால்
```

இங்கே மூன்று ஒப்பீடுகள் உள்ளன: மூலதனம் பூஜ்ஜியம், வட்டி_வீதம் பூஜ்ஜியம், வருடம் பூஜ்ஜியம். இவற்றின் இடையே “||” உள்ளது.

இதன் பொருள், மூலதனம் பூஜ்ஜியமாக இருக்கவேண்டும், அல்லது வட்டி_வீதம் பூஜ்ஜியமாக இருக்கவேண்டும், அல்லது வருடம் பூஜ்ஜியமாக இருக்கவேண்டும். இந்த மூன்றில் ஏதாவது ஒன்று சரியாக இருந்தாலும் கீழே தரப்பட்டுள்ள கட்டளைகள் நிறைவேறும்.

ஒன்றுக்கு மேற்பட்ட ஒப்பீடுகளைச் செய்யும்போது, அடைப்புக்குறிகளைச் சரியானமுறையில் பயன்படுத்தவேண்டும். இல்லாவிட்டால் சரியான பலன்கள் கிடைக்காது.

உதாரண நிரல்: leapyear.n

இதில் பின்வரும் வரியைக் கவனியுங்கள்:

```
@( ( ((வருடம் % 4) == 0) && ((வருடம் % 100) != 0) ) || ((வருடம் % 400) == 0) ) ஆனால்
```

இங்கே மூன்று ஒப்பீடுகள் உள்ளன. அவற்றை && மற்றும் || ஆகிய குறியீடுகளைப் பயன்படுத்தி, அடைப்புக்குறிகளின் உதவியுடன் இணைத்துள்ளோம். இப்படி:

“முதல் ஒப்பீடு” மற்றும் “இரண்டாம் ஒப்பீடு”) அல்லது “மூன்றாம் ஒப்பீடு”

இதன் பொருள், முதல் ஒப்பீடு, இரண்டாம் ஒப்பீடு இரண்டும் சரியாக இருக்கவேண்டும், அல்லது, மூன்றாம் ஒப்பீடுமட்டும் சரியாக இருக்கவேண்டும். அப்போது

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

கீழே தரப்பட்டுள்ள கட்டளைகள் நிறைவேறும்.

இதுபோல், ஒப்பீடுகளை நம் தேவைக்கேற்பப் பலவிதங்களில் இணைத்துப் பயன்படுத்த இயலும்.

10.9 இது வா? அல்லது, அது வா?

சில நேரங்களில், ஒரு குறிப்பிட்ட ஒப்பீடு உண்மை என்றால், நாம் சில கட்டளைகளை இயக்குவோம், ஒருவேளை அது உண்மை அல்ல என்றால், வேறு சில கட்டளைகளை இயக்குவோம். ஆங்கிலத்தில் இதனை If-Else Construct என்று அழைப்பார்கள்.

“எழில்” நிரலில் இதனை எப்படிச் செய்வது?

உதாரண நிரல்: [lcmgcd.n](#)

இதில் பின்வரும் வரிகளைக் கவனியுங்கள்:

```
@(எண்1 == எண்2) ஆனால்  
    <கட்டளை 1>  
@(எண்1 > எண்2) இல்லைஆனால்  
    <கட்டளை 2>  
இல்லை  
    <கட்டளை 3>  
முடி
```

இங்கே நாம் முதலில் எண்1, எண்2 இரண்டும் சமமா என்று பார்க்கிறோம், ஆம் எனில் ஒரு கட்டளையை நிறைவேற்றுகிறோம். ஒருவேளை அவை சமமாக இல்லாவிட்டால், எண்1 பெரியதா என்று பார்க்கிறோம், ஆம் எனில் இன்னொரு கட்டளையை நிறைவேற்றுகிறோம், அதுவும் இயலாவிட்டால் வேறொரு கட்டளையை நிறைவேற்றுகிறோம்.

இப்படி நாம் எத்துணை ஒப்பீடுகளை வேண்டுமானாலும் பிணைக்கலாம், அதற்கு ஓர் “ஆனால்” (IF), பல “இல்லைஆனால்”கள் (ElseIF), ஓர் “இல்லை” (Else) தேவைப்படும்.

10 -எழில் மொழி உதாரணங்கள் (Examples)

10.10 ஒரே பணியைத் திரும்பத் திரும்பச் செய்தல்

கணினியின் மிகப் பெரிய பலம், ஒரே பணியைத் திரும்பத் திரும்பப் பலமுறை விரைவாகச் செய்வதுதான். “எழில்” மொழியிலும் நீங்கள் அதனைச் சிறப்பாகச் செய்யலாம்.

இதில் இரண்டு வகைகள் உண்டு:

- **வரை:** ஒப்பீட்டைப் பரிசோதித்துவிட்டுக் கட்டளைகளை நிறைவேற்றுவது (While)
- **முடியேனில்:** கட்டளைகளை நிறைவேற்றிவிட்டு ஒப்பிடுவது (Until)

உதாரண நிரல்: tables.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
அ = 1
@ ( அ <= 10 ) வரை
    <கட்டளைகள்>
    அ = அ + 1
முடி
```

இங்கே “அ” என்ற மாறியின் மதிப்பு 1 ஆக உள்ளது. பின் அது படிப்படியாக அதிகரிக்கிறது, அது பத்து என்ற எண்ணைத் தாண்டும்வரை, சில கட்டளைகள் நிறைவேற்றப்படுகின்றன. “அ > 10” என்ற நிலை ஏற்பட்டதும் (அதாவது, “அ <= 10” என்ற ஒப்பீடு தவறானதும்), அந்தக் கட்டளைகள் நிறைவேற்றப்படுவது நின்றுவிடுகிறது.

“முடியேனில்” வாசகமும் கிட்டத்தட்ட இதேமாதிரிதான். ஆனால் கட்டளைகளை நிறைவேற்றியபிறகுதான் மாறியில் உள்ள மதிப்பு பரிசோதிக்கப்படும்.

உதாரண நிரல்: dowhile.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

அ = 0

செய்

<கட்டளைகள்>

அ = அ + 1

முடியேனில் @ (அ < 5)

முந்தைய நிரல் வரிகளோடு இதை ஒப்பிட்டுப் பாருங்கள், இரண்டும் கிட்டத்தட்ட ஒரேமாதிரி தோன்றினாலும், ஒரு முக்கியமான வித்தியாசம் உள்ளது.

அங்கேபோலவே இங்கேயும் “அ < 5” என்ற ஒப்பீடு இருக்கிறது, “அ” என்ற மாறியின் மதிப்பு ஒவ்வொன்றாக அதிகரிக்கிறது. ஆனால், அந்த ஒப்பீடு, கட்டளைகள் அனைத்தையும் நிறைவேற்றியபிறகுதான் நிகழ்த்தப்படுகிறது. இதுதான் இந்த இரு வாசகங்களுக்கும் இடையே முக்கியமான வேறுபாடு.

உங்களது நிரல்களில் “வரை” அல்லது “முடியேனில்” வாசகங்களைச் சரியானமுறையில் பயன்படுத்தினால் மிகப் பெரிய மதிப்புகளைக்கூடப் படிப்படியாக அலசி விடை காணமுடியும்.

8. ஒவ்வொன்றாக அலசுதல்

சில நேரங்களில் தொகுப்பான விவரங்களை ஒவ்வொன்றாக எடுத்து அலச வேண்டியுள்ளது. அப்போது உங்களுக்குப் பயன்படும் இரண்டு கட்டளைச் சொற்கள் “எழில்” மொழியில் உண்டு. அவை:

1. ஒவ்வொன்றாக (For Each)
2. ஆக (For)

உதாரண நிரல்: stringreverse.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

@ (சரம்1 இல் இ) ஒவ்வொன்றாக

சரம்2 = இ + சரம்2

முடி

இங்கே “சரம்1” என்பது ஓர் எழுத்துச் சரம். அதில் ஒவ்வொரு எழுத்தாக எடுத்து,

10 -எழில் மொழி உதாரணங்கள் (Examples)

“சரம்2” என்ற எழுத்துச் சரத்தில் மாற்றிச் சேமிக்கிறோம். அதற்கு “ஒவ்வொன்றாக” என்ற கட்டளை பயன்படுகிறது.

உதாரண நிரல்: [loopupdate.n](#)

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
@(எண் = 1, எண் <= 100, எண் = எண் + 1) ஆக
```

```
பதிப்பி எண்
```

```
முடி
```

இங்கே நாம் “எண்” என்ற மாறியில் 1ல் தொடங்கி ஒவ்வொன்றாக அதிகரித்து 100 வரை எண்களைச் சேமிக்கிறோம், அதைத் திரையில் பதிப்பிக்கிறோம்.

இதுபோல் எண்கள், எழுத்துகள் அடிப்படையிலான பல அலசல் பணிகளை இந்த இரு கட்டளைச் சொற்களைப் பயன்படுத்திச் சிறப்பாக எழுதலாம்.

குறிப்பாக, பட்டியல், dict என்ற இரு வகை விவரங்களை அலசுவதற்கு இது மிகவும் உபயோகப்படும். அதனைப் பின்னர் விரிவாகக் காண்போம்.

9. தேர்ந்தெடுத்தல்

ஒரே மாறியில் பல மதிப்புகள் இருக்கக்கூடும் என்ற சூழ்நிலையில், அவை ஒவ்வொன்றுக்கும் வெவ்வேறு கட்டளைகளை நிறைவேற்றுவதற்குத் “தேர்ந்தெடு” என்ற கட்டளைச் சொல் பயன்படும்.

உதாரண நிரல்: [select_case.n](#)

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
@(ஊர்) தேர்ந்தெடு
```

```
@("சென்னை") தேர்வு
```

```
<கட்டளை 1>
```

```
@("கோவை") தேர்வு
```

```
<கட்டளை 2>
```

```
@("மதுரை") தேர்வு
```

```
<கட்டளை 3>
```

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

ஏதேனில்

<கட்டளை 4>

முடி

இங்கே “ஊர்” என்ற மாறியில் உள்ள மதிப்பு சென்னையா, அல்லது கோவையா, அல்லது மதுரையா என்பதைப் பொறுத்து கட்டளை 1 அல்லது கட்டளை 2 அல்லது கட்டளை 3 நிறைவேற்றப்படும். இந்த மூன்றும் அல்லாத வேறு ஊர்களுக்குக் கட்டளை 4 நிறைவேற்றப்படும். இதைதான் “ஏதேனில்” என்ற சொல் குறிப்பிடுகிறது.

10.11 நிறுத்துதல், தொடர்தல்

நீங்கள் ஒரே செயலைப் பல மதிப்புகளுக்கு ஏற்ப திரும்பத் திரும்பச் செய்துகொண்டிருக்கும்போது, அவற்றை அவ்வப்போது நிறுத்தி அடுத்த மதிப்புக்குச் செல்லவேண்டியிருக்கலாம், அல்லது, அதனைச் செய்வதிலிருந்து வெளியேறவேண்டியிருக்கலாம். ஆங்கிலத்தில் இதனை Break, Continue என்பார்கள். இதற்கு இணையான “எழில்” மொழிச் சொற்கள் இவை:

1. நிறுத்து (break)
2. தொடர் (continue)

இதில் ‘நிறுத்து’ என்பது, நிறைவேற்றப்பட்டுக்கொண்டிருக்கும் தொடர்ச்சியான பணியை நிறுத்தி வெளியேறுதல், அதாவது, மீதமிருக்கும் மதிப்புகளையோ கட்டளைகளையோ நிறைவேற்றவேண்டியதில்லை.

உதாரண நிரல்: prime.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

@(p >= 2) வரை

@(fmod(n, p) == 0) ஆனால்

நிறுத்து

(முடி

p = p - 1

முடி

10 -எழில் மொழி உதாரணங்கள் (Examples)

இங்கே நாம் ஒரு “வரை” வளையத்தினுள் இருக்கிறோம். ஆனால் ஒரு குறிப்பிட்ட நிபந்தனை நிறைவேறும்போது, அடுத்தடுத்த எண்களைப் பரிசோதிக்காமல் வெளியேறிவிடுகிறோம், அதற்கு “நிறுத்து” என்ற கட்டளைச் சொல் பயன்படுகிறது.

அதாவது, ஒரு வளையம் 1, 2, 3 என்று 100 வரை செல்வதாக வைத்துக்கொள்வோம். அதை நீங்கள் ஏதோ காரணத்துக்காக 35 ல் நிறுத்த நேரலாம். அப்போது “நிறுத்து” என்ற சொல்லைப் பயன்படுத்தினால் 36, 37 முதலான எண்கள் பயன்படுத்தப்படாது.

உங்களது நிரல்களில் எப்போதெல்லாம் ஒரு வளையம் (Loop) இதற்குமேல் செயல்படவேண்டியதில்லை என்கிற சூழ்நிலை வருகிறதோ, அப்போதெல்லாம் இந்தக் கட்டளைச் சொல்லைப் பயன்படுத்துங்கள். இதன்மூலம் விடையும் சரியாகக் கிடைக்கும். கணினியின் செயல்திறனும் வீணாகாமல் காக்கப்படும்.

அடுத்து, “தொடர்” என்ற கட்டளைச் சொல். இது “நிறுத்து”க்கு நேர் எதிரானது. இங்கே நீங்கள் வளையத்தைத் தொடர்ந்து செயல்படுத்த விரும்புகிறீர்கள். ஆனால் இந்தக் குறிப்பிட்ட மதிப்புக்கு மீதமிருக்கும் கட்டளைகளைப் பயன்படுத்த விரும்பவில்லை.

அதாவது, வளையம் 1, 2, 3 என்று தொடர்ந்து 100 வரை செல்கிறது. அதே 35 ல் “நிறுத்து”க்குப் பதில் “தொடர்” என்ற வாசகத்தைப் பயன்படுத்துகிறோம். அப்போது, 35 என்ற மதிப்புக்கு மீதமுள்ள கட்டளைகள் நிறைவேற்றப்படாது. ஆனால் 36 முதல் உள்ள கட்டளைகள் எப்போதும்போல் நிறைவேற்றப்படும்.

உதாரண நிரல்: nqueens.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
@( i = 0, i < n, i = i + 1 ) ஆக
    <சில கட்டளைகள்>
    @( j < col ) ஆனால்
        தொடர்
    முடி
    <மேலும் சில கட்டளைகள்>
முடி
```

இங்கே “சில கட்டளைகள்” முதலில் நிறைவேற்றப்படும், அதன்பிறகு ஓர் ஒப்பீடு (“j < col”) செய்யப்படும். ஒருவேளை அது நிஜம் எனில், மீதமுள்ள “மேலும் சில கட்டளைகள்” இந்த மதிப்புக்கு நிறைவேற்றப்படாது, “ஆக” வளையம் அடுத்த மதிப்புக்குச்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

சென்றுவிடும்.

"நிறுத்து" மற்றும் "தொடர்" இடையிலான வித்தியாசம் நுட்பமானது. அதை அறிந்து சரியாகப் பயன்படுத்தினால் உங்கள் நிரல் மிகத் துல்லியமாக இயங்கும்.

10.12 அதிக விவரங்கள்

"எழில்" மொழியில் அதிக விவரங்களைத் தொகுத்துவைக்க இரண்டு வழிகள் உண்டு:

1. பட்டியல்
2. Dict

இதில் பட்டியல் 1D எனப்படும் ஒற்றைப் பரிமாண வகையைச் சேர்ந்தது. அதாவது, ஒன்றன்மீது ஒன்றாகப் பல விவரங்களைத் தொகுத்துவைக்கலாம். உதாரணமாக, மாணவர்களின் பெயர்கள்.

அதே பட்டியலில், ஒவ்வொரு மாணவர் பெயருக்கும் அருகே அவர் வாங்கிய மதிப்பெண்களைச் சேர்க்க விரும்பினால், அதற்கு Dict பயன்படும். இது 2D, இரட்டைப் பரிமாண வகையைச் சேர்ந்தது.

பட்டியல், Dict இரண்டிலும் நாம் விவரங்களைத் தொகுக்கலாம், எடுக்கலாம். அதற்கான வழிவகைகள் அனைத்தும் தரப்பட்டுள்ளன.

உதாரண நிரல்: string_demo.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
ப = பட்டியல்()
பின்இணை( ப , "வாழை")
பின்இணை( ப , "கொய்யா" )
பின்இணை( ப , "மா")
```

```
பதிப்பி எடு(ப,0), எடு(ப ,1)
```

10 -எழில் மொழி உதாரணங்கள் (Examples)

தலைகீழ்(ப)

வரிசைப்படுத்து(ப)

பதிப்பி "பட்டியல் நீளம்", நீளம்(ப)

நீட்டிக்க(ப , ப)

இந்த வரிகளில் நாம் பட்டியலை உருவாக்குவதை அறிகிறோம், அதில் விவரங்களைச் சேர்க்கத் தெரிந்துகொள்கிறோம், அவற்றை எடுத்துத் திரையில் அச்சிடப் பழகுகிறோம், தலைகீழாக்குகிறோம், வரிசைப்படுத்துகிறோம், நீளம் கண்டுபிடிக்கிறோம், நீட்டிக்கிறோம்...

இப்படி உங்களுடைய நிரல்களிலும் விவரங்களைச் சேர்த்துப் பயன்படுத்தும் தேவை வரும்போதெல்லாம், பட்டியலைப் பயன்படுத்திக்கொள்ளுங்கள்.

உதாரண நிரல்: dict.n

இதில் இந்த வரிகளைக் கவனியுங்கள்:

```
z = dict()
```

```
வை(z,"name","மணி")
```

```
வை(z,"age",10)
```

```
வை(z,"location","தஞ்சாவூர்")
```

```
பதிப்பி "என் பெயர் "+எடு(z,"name")
```

```
பதிப்பி "நான் வாழும் ஊர் "+எடு(z,"location")
```

```
பதிப்பி "எனக்கு "+str(எடு(z,"age"))+" வயதாகிறது. "
```

இதை முந்தின உதாரணத்துடன் ஒப்பிட்டுப் பாருங்கள், மணி என்பவருடைய வயது, இருப்பிடம் என்பதுபோன்ற பல விவரங்களை இங்கே சேமிக்கிறோம். ஆனால் "பட்டியல்" வகையில் மணி, சுரேஷ், ரமேஷ் போன்ற பெயர்களைமட்டும் சேமிக்கலாம், அல்லது 10, 12, 25 என்று வயதுகளைமட்டும் சேமிக்கலாம், இப்படித் தொகுக்க இயலாது.

"பட்டியல்" போதும் என்கிற இடங்களில் "Dict" பயன்படுத்துவது வீண். அதேசமயம் "Dict" தேவைப்படும் இடங்களில் அந்தப் பணியைப் "பட்டியல்"ஆல் செய்ய முடியாது. இந்த வித்தியாசத்தை உணர்ந்து உங்கள் நிரல்களை வடிவமைப்பது நல்லது.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

10.13 நிரல்பாகங்கள்

ஒரேமாதிரி பணியை ஒரு நிரலில் பல இடங்களில் செய்யவேண்டியிருந்தால், Function எனப்படும் கட்டமைப்பைப் பயன்படுத்துவார்கள். அதாவது, ஒருமுறை எழுதிவிட்டு அதனைப் பலமுறை அழைத்துப் பயன்படுத்துவது.

இங்கே "எழில்" மொழியிலும் Functions உண்டு. அவற்றை "நிரல்பாகம்" என்று நாம் அழைக்கிறோம்.

உதாரண நிரல்: [convert2kelvin.n](#)

இதில் இந்த வரிகளைக் கவனியுங்கள்:

நிரல்பாகம் கெல்வின்_இருந்து_மாற்று(k)

<கட்டளைகள்

முடி

கெல்வின்_இருந்து_மாற்று(0)

இங்கே நாம் ஒரு நிரல்பாகத்தை எழுதி, அதனை உரிய இடத்திலிருந்து அழைக்கிறோம். அந்தக் கட்டளைகள் கச்சிதமாக நிறைவேற்றப்படுகின்றன. திரும்பத் திரும்ப ஒரே பணியை வெவ்வேறு இடங்களில் செய்யவேண்டிய தேவை ஏற்படும்போது, நிரல்பாகம் மிகவும் பயன்படும்.

11 - நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

எழில் குறிப்புகள் மற்றும் நிரலாளர் உதவிக் கையேடு. இந்த அத்தியாயத்தில் உயர்நிலை செயல்பாடுகளை கானலாம். இதை உடனே புறிந்து கொள்ள வேண்டும் என்ற அவசியமில்லை. பல முறை வாசிக்க அழைக்கிறேன்.

11.1 பதிப்பி

பயன்படுத்தும் விதம் ::= "பதிப்பி" எழுத்துச்சரம் அல்லது மாறியின் பெயர்
["", " எழுத்துச்சரம் அல்லது மாறியின் பெயர்]

இந்தக் கட்டளை தன்னிடம் தரப்படுகிற எழுத்துச்சரம் அல்லது மாறியின் மதிப்பைத் திரையில் காண்பிக்கிறது. எழுத்துச்சரம் மற்றும் மாறியின் பெயர் இரண்டும் கலந்து தரப்பட்டால், அவை ஒன்று சேர்க்கப்பட்டு (காண்க: எழுத்துச்சரங்களை இணைத்தல்) காண்பிக்கப்படும்.

பொதுவாக இந்தக் கட்டளை இரண்டுவிதங்களில் பயன்படும்: ஒன்று, பயனாளர்களுக்கு ஒரு நிரல் இயங்கும் தன்மையை, இடையில் உள்ள முன்னேற்றங்களைத் தெரிவிக்க. இரண்டாவது, ஏதேனும் பிழைகள் நேர்ந்தால் அதனைப் பயனாளருக்கும் நிரல் எழுதுவோருக்கும் தெரிவிக்க.

இந்தக் கட்டளையில் தரப்படும் எழுத்துச்சரங்கள் மற்றும் மாறிகள் அனைத்தும் இணைந்து ஒரு வரியில் காண்பிக்கப்படும், அத்துடன் அந்த வரி முழுமை பெறும். அடுத்த கட்டளை அடுத்த வரியில்தான் அச்சாகும். இதன் பொருள், ஒவ்வொரு "பதிப்பி" கட்டளையின் நிறைவிலும் "\n" தானே சேர்க்கப்படுகிறது, நீங்கள் அதைத் தனியே சேர்க்கவேண்டியதில்லை.

11.2 உள்ளீடு

பயன்படுத்தும் விதம் ::= "உள்ளீடு(" எழுத்துச்சரம் ")"

இந்தக் கட்டளை இருவிதமாக இயங்குகிறது. தன்னிடம் தரப்படுகிற எழுத்துச் சரத்தைத் திரையில் காண்பிக்கிறது. பின்னர், பயனாளர் தட்டச்சு செய்யப்போகும் விவரத்துக்காகக் காத்திருக்கிறது. அவர் தட்டச்சு செய்து "Enter" விசையைத் தட்டியவுடன், அந்த விவரத்தை நிரலுக்குத் தருகிறது.

பயனாளர்கள் இந்தக் கட்டளைக்குப் பதிலாக ஓர் எண்ணையோ எழுத்துகளையோ தேதியையோ தரலாம். அவை அனைத்தும் எழுத்து வடிவில்தான் நிரலுக்குத் தரப்படும். வேண்டுமானால் எழுத்தை எண்ணாக, தேதியாக மாற்றிக்கொள்ளவேண்டிய **Typcasting** பொறுப்பு நிரல் எழுதுவோருடையது.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

உதாரணமாக, “உங்கள் வயது என்ன?” என்ற கேள்விக்குப் பயனாளர் “10” என்று பதில் அளித்தால், அதை நிரலின் அடுத்த வரியில் அப்படியே பயன்படுத்த இயலாது. Int எனப்படும் எண் வகைக்கு மாற்றிதான் பயன்படுத்தவேண்டும். இப்படி:

வயது = int(உள்ளீடு(“உங்கள் வயது என்ன?”))

11.3 ஆனால்

பயன்படுத்தும் விதம் ::= “@” நிபந்தனை வாசகம் நிபந்தனை வாசகங்கள் “)”
ஆனால்

செயல் வாசகங்கள்

”முடி”

நிபந்தனை வாசகம் ::= மாறியின் பெயர் அல்லது மதிப்பு “ ” தர்க்கக் குறியீடு
” ” மாறியின் பெயர் அல்லது மதிப்பு

அனுமதிக்கப்பட்டுள்ள தர்க்கக் குறியீடுகள் ::= == | != | > | < | >= | <=

நிபந்தனை வாசகங்கள் ::= “(” நிபந்தனை வாசகம் “)” “!”&& “(” அல்லது “||” “(”
நிபந்தனை வாசகம் “)”

இந்தக் கட்டளை நீங்கள் தருகிற நிபந்தனை வாசகம் சரியா, தவறா என்பதைக் குறிப்பிடுகிறது. ஒருவேளை நிபந்தனை வாசகம் சரியாக இருந்தால், “ஆனால்”, “முடி” என்ற கட்டளைகளுக்கு இடையே உள்ள கட்டளைகள் அனைத்தும் நிறைவேற்றப்படும். இல்லாவிட்டால் நிரலின் கட்டுப்பாடு “முடி” என்ற சொல்லுக்குப் பின்னர் உள்ள கட்டளைக்குச் சென்றுவிடும்.

இதில் ஒன்றுக்கு மேற்பட்ட நிபந்தனை வாசகங்களையும் தரலாம். அப்போது அவற்றை “&&” (மற்றும்), “||” (அல்லது) என்ற குறியீடுகளில் ஒன்றால் பிணைக்கவேண்டும்.

ஒருவேளை இரண்டுக்கு மேற்பட்ட நிபந்தனை வாசகங்களை ஒரே இடத்தில் தொகுத்தால், அப்போது () என்ற அடைப்புக்குறியைக் கவனமாகப் பயன்படுத்துங்கள். இவை கணிதச் சமன்பாடுகள் நிறைவேற்றப்படும் அதே தன்மைப்படி நிறைவேற்றப்படும். உதாரணமாக:

@(((a>5) || (b<5)) && (c>5)) ஆனால்

11.4. இல்லை ஆனால் & இல்லை

பயன்படுத்தும் விதம் ::= “@” நிபந்தனை வாசகம் அல்லது வாசகங்கள் “)”
ஆனால்

11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

```
        செயல் வாசகங்கள்
    ["@(" நிபந்தனை வாசகம் அல்லது வாசகங்கள் ") இல்லைஆனால்"
        செயல் வாசகங்கள்
        "இல்லை"
        செயல் வாசகங்கள்]
    "முடி"
```

இந்தக் கட்டளை எப்போதும் “ஆனால்” என்ற கட்டளையுடன் இணைந்தே இயங்கும். “ஆனால்”க்கு முன்பாகத் தரப்பட்டுள்ள நிபந்தனை சரி என்றால், அதை அடுத்துத் தரப்பட்டுள்ள செயல் வாசகங்கள் இயங்கும், ஒருவேளை அது தவறு என்றால், “இல்லைஆனால்”க்கு முன்பே தரப்பட்டுள்ள அடுத்த நிபந்தனை வாசகம் செயல்படுத்தப்படும்.

இப்படி வரிசையாக வரும் நிபந்தனை வாசகங்களில் எவையும் சரி இல்லை என்றால், நிறைவாக “இல்லை” என்ற சொல் தரப்பட்டிருக்கும். அதை அடுத்து வருகிற செயல் வாசகங்கள் இப்போது நிறைவேற்றப்படும்.

“ஆனால்”, “இல்லைஆனால்”, “இல்லை” ஆகிய மூன்றுக்கும் இணையாக “முடி” என்ற சொற்றொடர் வரும். இதுவே {} என்ற அடைப்புக்குறி இணையைப்போல் இங்கே செயல்படுகிறது.

11.5. வரை

```
        பயன்படுத்தும் விதம் ::= "@(" நிபந்தனை வாசகம் அல்லது வாசகங்கள் ")
    வரை"
        செயல் வாசகங்கள்
    "முடி"
```

இந்தக் கட்டளை சுழல் வகையைச் சேர்ந்ததாகும். அதாவது, இதனுள் தரப்பட்டுள்ள செயல் வாசகங்கள் ஒரு முறையோ அல்லது அதற்கு மேலோ இயங்கக்கூடும். சில நேரங்களில் அவை இயங்காமலே போகும் வாய்ப்பும் உண்டு.

“ஆனால்” கட்டளையைப்போலவே இங்கேயும் நிபந்தனை வாக்கியங்கள் உள்ளன. அவை ஒன்றுக்கு மேற்பட்டமுறை பரிசோதிக்கப்படும். எப்போது அந்த நிபந்தனை சரியாக உள்ளதோ, அப்போதெல்லாம் செயல் வாசகங்கள் இயக்கப்படும். நிபந்தனை தவறாகும்போது, இந்தச் சுழல் நிறைவடைந்து அடுத்த வாசகத்துக்குச் சென்றுவிடும்.

இதில் ஒன்றுக்கு மேற்பட்ட நிபந்தனைகளைச் சேர்க்கலாம். அதற்கு “&&” அல்லது “||” என்ற சொற்களையும், () அடைப்புக்குறிகளையும் உரிய முறையில் பயன்படுத்தவேண்டும்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

11.6. செய் & முடியேனில்

பயன்படுத்தும் விதம் ::= “செய்”

செயல் வாசகங்கள்

”முடியேனில் @(" நிபந்தனை வாசகம் அல்லது வாசகங்கள் ")”

இந்தக் கட்டளையும் “வரை” போன்றதே. ஆனால் ஒரு முக்கியமான வித்தியாசம், இங்கே செயல் வாசகங்கள் நிறைவேற்றப்பட்டபிறகுதான் நிபந்தனை வாசகமோ வாசகங்களோ நிறைவேற்றப்படும்.

மற்றபடி பயன்படுத்தும் விதம் இவை இரண்டுக்கும் ஒன்றே. இதில் உள்ள செயல் வாசகங்கள் ஒருமுறையாவது நிச்சயம் செயல்படுத்தப்படும் என்பதை மறந்துவிடவேண்டாம். எப்போது “வரை”யைப் பயன்படுத்துவது, எப்போது “முடியேனில்”ஐப் பயன்படுத்துவது என்று தீர்மானிக்க உதவும் அம்சம் இதுதான்.

11.7. ஒவ்வொன்றாக

பயன்படுத்தும் விதம் ::= “@(" தொகுப்பு மாறியின் பெயர் “ இல் “ தனி

மாறியின் பெயர் ") ஒவ்வொன்றாக”

செயல் வாசகங்கள்

”முடி”

அனுமதிக்கப்பட்டுள்ள தொகுப்பு மாறி வகைகள் ::= பட்டியல் | dict |

எழுத்துச்சரம்

இந்தக் கட்டளை தொகுப்பாக உள்ள விஷயங்களை ஒவ்வொன்றாக எடுத்துச் செயல்படுத்துகிறது. உதாரணமாக, ஓர் எழுத்துச் சரத்தில் உள்ள எழுத்துகளை ஒவ்வொன்றாகக் கையாள்வது; ஒரு பட்டியல் அல்லது Dict வகையில் உள்ள விஷயங்களை ஒவ்வொன்றாகக் கையாள்வது போன்றவை.

இங்குள்ள தொகுப்பில் எத்தனை அம்சங்கள் உள்ளனவோ, அத்தனைமுறை இதில் உள்ள செயல் வாசகங்கள் இயங்கும். உதாரணமாக, ஒரு பட்டியலில் 10 பெயர்கள் இருப்பின், செயல் வாசகங்கள் பத்து முறை இயங்கும்.

ஒருவேளை நீங்கள் இந்தச் சுழலில் இருந்து பாதியில் வெளியேற விரும்பினால், உங்கள் தேவையைப் பொறுத்து “நிறுத்து” அல்லது “தொடர்” என்ற வாசகங்களைப் பயன்படுத்தலாம்.

11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

11.8. ஆக

பயன்படுத்தும் விதம் ::= "@" தொடக்க நிலை ", " நிபந்தனை நிலை ", " அதிகரிப்பு நிலை ") ஆக
செயல் வாசகங்கள்
"முடி"

தொடக்க நிலை ::= மாறியின் பெயர் அல்லது மதிப்பு " = " மாறியின் பெயர் அல்லது மதிப்பு

நிபந்தனை நிலை ::= மாறியின் பெயர் அல்லது மதிப்பு " " தர்க்கக் குறியீடு " " மாறியின் பெயர் அல்லது மதிப்பு

அனுமதிக்கப்பட்டுள்ள தர்க்கக் குறியீடுகள் ::= == | != | > | < | >= | <=

அதிகரிப்பு நிலை ::= மாறியின் பெயர் " = " மாறியின் பெயர் " கணிதக் குறியீடு " " மாறியின் பெயர் அல்லது மதிப்பு

அனுமதிக்கப்பட்டுள்ள கணிதக் குறியீடுகள் ::= + | - | * | /

இந்தக் கட்டளையில் மூன்றுவிதமான வாசகங்கள் பயன்படுத்தப்படுகின்றன: தொடக்க நிலை, நிபந்தனை நிலை, அதிகரிப்பு நிலை.

இவற்றில் தொடக்க நிலை என்பது முதலில் நிறைவேற்றப்படும் வாசகம். பின்னர், அதிகரிப்பு நிலை என்பது அதன்மீது செயல்படுத்தப்படும். நிறைவாக, நிபந்தனை நிலை எட்டப்பட்டுவிட்டதா என்பது காணப்படும்.

உதாரணமாக, @(a=1;a<10;a=a+1) என்று குறிப்பிடப்பட்டிருந்தால், முதலில் தொடக்க நிலையில் "a" என்ற மாறியில் "1" என்ற மதிப்பு வைக்கப்படுகிறது. பின் அது ஒவ்வொன்றாக அதிகரிக்கப்படுகிறது, "a" மதிப்பு 10 ஐவிடக் குறைவாக உள்ளவரை செயல் வாசகங்கள் நிறைவேற்றப்படும். அதன்பிறகு, இந்தச் சுழல் தானாக நிறைவுக்கு வந்துவிடும்.

"ஆக" சுழல் எத்தனைமுறை நிறைவேற்றப்படும் என்பதைத் தீர்மானிப்பது இந்த மூன்று நிலை வாசகங்கள்தான். இங்கேயும் "நிறுத்து", "தொடர்" என்ற வாசகங்களைப் பயன்படுத்திச் சுழலின் தன்மையை மாற்றி அமைக்கலாம்.

11.9. தேர்ந்தெடு, தேர்வு & ஏதேனில்

பயன்படுத்தும் விதம் ::= "@" மாறியின் பெயர் ") தேர்ந்தெடு"
"@" மாறியின் மதிப்பு ") தேர்வு"

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

செயல் வாசகங்கள்
["@(" மாறியின் மதிப்பு ") தேர்வு"
செயல் வாசகங்கள்
"ஏதேனில்"
செயல் வாசகங்கள்]
"முடி"

இந்தக் கட்டளையில் ஒரு மாறியின் மதிப்பு காணப்படுகிறது. அதன் மதிப்பு வெவ்வேறு எண்களாகவோ எழுத்துகளாகவோ இருப்பின், அதற்குரிய "தேர்வு" சொற்றொடருக்குக் கீழே உள்ள செயல் வாசகங்கள் நிறைவேற்றப்படுகின்றன.

ஒருவேளை, அதன் நிஜமான மதிப்பு அங்கே தரப்பட்டுள்ள எந்த மதிப்போடும் பொருந்தவில்லை என்றால், "ஏதேனில்"க்குக் கீழே உள்ள செயல் வாசகங்கள் நிறைவேற்றப்படுகின்றன.

இங்கே மாறியின் மதிப்பு ஒரே ஒருமுறைதான் பரிசோதிக்கப்படும் என்பதை நினைவில் கொள்க. இது பலமுறை இயங்கும் சுழல் அல்ல.

11.10 நிறுத்து

பயன்படுத்தும் விதம் ::= "நிறுத்து"

இந்தக் கட்டளை எப்போதும் ஒரு சுழலின் நடுவே பயன்படுத்தப்படும். அத்துடன் அந்தச் சுழல் நிறைவடைந்துவிடும்.

அதாவது, சுழல் வழக்கம்போல் நிறைவேற இன்னும் பல சுற்றுகள் மீதமிருந்தாலும், இந்தக் கட்டளை அதனை அப்படியே நிறுத்திவிடும். பிழையான மதிப்புகள், நாம் செய்ய விரும்பியதைச் செய்து முடித்துவிட்டோம் என்கிற சூழல்களில் இந்தக் கட்டளையைப் பயன்படுத்தலாம். கட்டளை உள்ள உள் சுழல் தானாக நிறைவடைந்துவிடும். ஒருவேளை ஒன்றுக்கு மேற்பட்ட சுழல்கள் இருப்பின், அவை அனைத்திலிருந்தும் வெளியேற இன்னும் சில "நிறுத்து" கட்டளைகளை உரிய இடங்களில் பயன்படுத்தவேண்டியிருக்கலாம்.

11.11 தொடர்

பயன்படுத்தும் விதம் ::= "தொடர்"

இந்தக் கட்டளையும் "நிறுத்து" போன்றதேதான். ஆனால் இது சுழலில் இருந்து மொத்தமாக வெளியேறுவதில்லை. அந்தக் குறிப்பிட்ட சுற்றைமட்டும் நிறுத்திவிடுகிறது.

உதாரணமாக, ஒரு சுழலில் 10 படிகள் இருக்கலாம், அவை 1, 2, 3, 4, 5 என்று

11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

வரும்போது நாம் “நிறுத்து” கட்டளையை அழைத்தால், 5, 6, 7, 8, 9, 10 ஆகிய சுற்றுகள் இயக்கப்படாது. அதே இடத்தில் “தொடர்” கட்டளையை அழைத்தால், 5 ம் சுற்றும்மட்டும் இயங்காது. 6 ல் தொடங்கி மற்ற சுற்றுகள் எப்போதும்போல் இயங்கும்.

இந்த வித்தியாசத்தைப் புரிந்துகொண்டுவிட்டால், எப்போது “நிறுத்து”, எப்போது “தொடர்” என்பது தெளிவாக விளங்கிவிடும்.

11.12 பட்டியல்

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் “ = பட்டியல்0”

இந்தக் கட்டளை ஒரு புதிய தொகுப்பினை உருவாக்குகிறது. அதன் வகை “பட்டியல்” என்பதாகும்.

ஒரு “பட்டியல்” மாறியில் எண், எழுத்து, தேதி போன்ற பல அம்சங்களைச் சேமிக்கலாம், வேண்டியபோது அவற்றை மீட்டெடுக்கலாம். அதற்கான கட்டளைகள் கீழே தரப்பட்டுள்ளன.

11.13 பின்இணை

பயன்படுத்தும் விதம் ::= “பின்இணை(” பட்டியல் வகை மாறியின் பெயர் “,” இணைக்கப்படவேண்டிய மதிப்பு “)”

அனுமதிக்கப்பட்டுள்ள மதிப்பு வகைகள் ::= எண் | எழுத்து | எழுத்துச்சரம் | தேதி

இந்தக் கட்டளை ஒரு பட்டியல் வகை மாறியின் பின்னால் ஒரு மதிப்பைச் சேர்க்க உதவுகிறது. உதாரணமாக, ஏற்கெனவே ஐந்து விஷயங்களைக் கொண்ட பட்டியலில் ஆறாவதாக ஒன்றைச் சேர்க்க விரும்பினால், இந்தக் கட்டளையைப் பயன்படுத்தலாம். எண்கள், எழுத்துகள், எழுத்துச் சரங்கள், தேதி போன்றவற்றை இதன்மூலம் பட்டியலில் சேர்க்கலாம்.

11.14 எடு

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் “ = எடு(” பட்டியல் வகை மாறியின் பெயர் “,” எண் வரிசை “)”

இந்தக் கட்டளை ஒரு பட்டியலில் உள்ள விஷயங்களை மீட்டெடுத்துப் பயன்படுத்த உதவுகிறது. நாம் சேர்த்த வரிசைப்படி விஷயங்களை மீட்கலாம். அதற்கான வரிசை எண்ணைக் குறிப்பிட்டாலே போதுமானது.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

ஒருவேளை அந்த வரிசை எண் பட்டியலில் இல்லை என்றால், அதற்கான பிழைச் செய்தி தரப்படும். இல்லாவிட்டால் அந்த வரிசை எண்ணில் சேமிக்கப்பட்டுள்ள விவரம் மீட்டுத் தரப்படும்.

11.15 தலைகீழ்

பயன்படுத்தும் விதம் ::= "தலைகீழ்(" பட்டியல் வகை மாறியின் பெயர் ")"

இந்தக் கட்டளை ஒரு பட்டியலைத் தலைகீழாக மாற்றிவிடுகிறது. அதாவது, நாம் அதில் விவரங்களைச் சேர்த்த வரிசைக்கு நேர் எதிரானதாக. நிறைவாகச் சேர்த்த விஷயம் முதலாவதாக வரும், அதற்குச் சற்றுமுன் சேர்த்த விஷயம் இரண்டாவதாக... இப்படித் தொடர்ந்து முதலாவதாகச் சேர்த்த விஷயம் நிறைவாக நிற்கும்.

ஆனால், இதனால் விவரங்களோ, பட்டியலின் நீளமோ மாறாது. வரிசை எண்கள்மட்டுமே மாறும்.

11.16 வரிசைப்படுத்து

பயன்படுத்தும் விதம் ::= "வரிசைப்படுத்து(" பட்டியல் வகை மாறியின் பெயர் ")"

இந்தக் கட்டளை ஒரு பட்டியலில் உள்ள விவரங்களை அகர வரிசைப்படி (அல்லது, ஏறு வரிசைப்படி) அடுக்கித் தரும். உதாரணமாக, பட்டியலில் மாணவர்களின் பெயர்கள் இருந்தால், அவை அகரவரிசையில் அடுக்கப்படும். மாணவர்களின் மதிப்பெண்கள் இருந்தால், அவை ஏறுவரிசையில் அடுக்கப்படும்.

இம்முறையும், பட்டியலின் நீளம் மாறாது. புதிய விஷயங்கள் சேர்க்கப்படாது. வரிசை எண்மட்டுமே மாறும்.

11.17 நீளம்

பயன்படுத்தும் விதம் ::= எண் வகை மாறியின் பெயர் " = நீளம்(" பட்டியல் வகை மாறியின் பெயர் ")"

இந்தக் கட்டளை ஒரு பட்டியலின் நீளத்தை எண்ணாகத் தரும். இதனைப் பயன்படுத்திக் கணக்கீடுகளைச் செய்யலாம்.

11.18 நீட்டிக்க

11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

பயன்படுத்தும் விதம் ::= "நீட்டிக்க(" பட்டியல் வகை மாறியின் பெயர் " , " பட்டியல் வகை மாறியின் பெயர் ")"

இந்தக் கட்டளை ஒரு வகைப் பட்டியலுடன் இன்னொரு வகைப் பட்டியலை ஒட்டி நீட்டித்துத் தரும். அதாவது, முதலில் தரப்பட்டிருக்கும் பட்டியலின் நிறைவில் இரண்டாவதாகத் தரப்பட்டிருக்கும் பட்டியல் இணைக்கப்பட்டு, அந்தப் பெரிய பட்டியல் முதல் பட்டியலின் மாறியில் சேமிக்கப்படும்.

இதுவும் "பின்இணை" கட்டளையும் கிட்டத்தட்ட ஒரேமாதிரியானதுதான். ஆனால் அங்கே ஒரே ஒரு விஷயத்தைதான் சேர்க்க இயலும், இங்கே முழுப் பட்டியலையும் சேர்க்க இயலும். அதுதான் முக்கியமான வித்தியாசம்.

11.19 dict

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் " = dict()

இந்தக் கட்டளை "பட்டியல்" போன்றதுதான். ஆனால் மாறியின் வகை "Dict" என பயன்படுத்தவேண்டும். இதில் தனித்தனி அம்சங்களை அன்றி, ஒவ்வொன்றையும் ஓர் அடையாளத்துடன் சேமிக்க இயலும்.

உதாரணமாக, ஒரு மாணவரின் பெயர், வயது, முகவரி போன்றவற்றை ஒரே "Dict" மாறியில் சேமிக்கலாம், வேண்டியபோது அவற்றை மீட்டு எடுக்கலாம்.

11.20 வை

பயன்படுத்தும் விதம் ::= "வை(" dict வகை மாறியின் பெயர் " , " எழுத்துச்சரம் " , " மாறியின் பெயர் அல்லது மதிப்பு ")"

இந்தக் கட்டளை ஒரு "Dict" வகை மாறியில் ஒரு புதிய மதிப்பைச் சேர்க்கிறது. அதனைப் "பட்டியல்"போல் வெறுமனே சேர்க்காமல், எழுத்துச்சரமான ஓர் அடையாளத்தைச் சேர்த்துப் பதிவு செய்கிறோம்.

உதாரணமாக, "பெயர்" என்ற அடையாளத்துடன் "ராமு" என்ற மதிப்பைச் சேர்க்கலாம்; பின் "வயது" என்ற அடையாளத்துடன் "12" என்ற மதிப்பைச் சேர்க்கலாம்... இப்படியே நமக்கு வேண்டிய எல்லா விவரங்களையும் இந்தக் கட்டளையை மீண்டும் மீண்டும் அழைப்பதன்மூலம் சேமிக்கலாம்.

11.21 எடு

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் " = எடு(" dict வகை மாறியின்

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

பெயர் “ , “ எழுத்துச்சரம் “)”

இந்தக் கட்டளை “Dict”ல் வைத்த ஒரு விஷயத்தை எடுக்க உதவுகிறது. அதன் அடையாளம் என்ன என்பதை எழுத்துச்சரமாகக் குறிப்பிட்டால், அதில் பதிவு செய்யப்பட்ட மதிப்பு மீட்டுத் தரப்படும்.

ஒருவேளை அந்த அடையாளத்தில் எந்த மதிப்பும் பதிவு செய்யப்படவில்லை எனில், பிழைச்செய்தி காண்பிக்கப்படும்.

11.22 நிரல்பாகம்

பயன்படுத்தும் விதம் ::= “நிரல்பாகம் “ எழுத்துச்சரம் “ (” [மாறியின் பெயர் [” , ” மாறியின் பெயர்]] “)”
செயல் வாசகங்கள்
”முடி”

இந்தக் கட்டளை ஒரு நிரலில் அடிக்கடி பயன்படும் நிரல் வரிகளைத் தனியே நிரல்பாகமாகச் சேமிக்க உதவுகிறது. பின்னர் மீதமுள்ள நிரலில் வேண்டும்போதெல்லாம் நமக்குத் தேவையான நிரல்பாகத்தை அழைத்துப் பயன்படுத்திக்கொள்ளலாம்.

ஒவ்வொரு நிரல்பாகத்திலும் நாம் மாறிகளை உள்ளீடு செய்யலாம் (Input). இவை பூஜ்ஜியம் முதல் எத்தனையாக வேண்டுமானாலும் இருக்கலாம். நிரல்பாகத்தை அழைக்கும்போது, இந்த மாறிகளுக்கு உரிய மதிப்பை இட்டு அழைக்கவேண்டும்.

11.23 பின்கொடு

பயன்படுத்தும் விதம் ::= “பின்கொடு “ மாறியின் பெயர் அல்லது மதிப்பு

இந்தக் கட்டளை ஒரு நிரல்பாகத்தின் விடையாக ஏதேனும் ஒரு மதிப்பைத் திரும்ப அளிப்பதைக் குறிப்பிடுகிறது. இந்த மதிப்பு எண்ணாகவோ, எழுத்தாகவோ, எழுத்துச் சரமாகவோ, பட்டியலாகவோ, வேறு வகை மாறியாகவோ இருக்கலாம்.

ஒவ்வொரு நிரல்பாகமும் ஒரே ஒரு மதிப்பைமட்டும்தான் பின்கொடுக்க இயலும். ஆகவே, ஒவ்வொரு நிரல்பாகத்திலும் இந்த வரிக்குப் பிறகு ஒருவேளை வேறு கட்டளைகள் எவையேனும் இருந்தால் அவை நிறைவேற்றப்படாது என்பதைக் கவனத்தில் கொள்ளவேண்டும்.

11.24 கோப்பை_திற

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் “ = கோப்பை_திற(” எழுத்துச்சரம்

11 -நிரலாளர் உதவிக் கையேடு (Programmer's Examples)

[", " எழுத்து] ")

அனுமதிக்கப்பட்டுள்ள எழுத்து வகைகள் = "w"

இந்தக் கட்டளை கணினியில் உள்ள கோப்பு ஒன்றைத் திறக்கப் பயன்படுத்தப்படுகிறது. அதற்கான முழு முகவரி ("எழில்" மென்பொருள் நிறுவப்பட்டுள்ள இடத்திலிருந்து) தரப்படவேண்டும். கோப்பைத் திறக்கும் விதம் வெறுமனே வாசிப்பதாக இருக்கலாம், அல்லது எழுதுவதற்காக இருக்கலாம்.

எழுதுவதற்காக ஒரு கோப்பைத் திறக்க முயற்சி செய்யும் நேரத்தில், ஒருவேளை அந்தக் கோப்பு அந்த இடத்தில் இல்லாவிட்டால், புதிய கோப்பு ஒன்று உருவாக்கப்படும், அதன்பிறகு திறக்கப்படும்.

ஒருவேளை கோப்பைத் திறக்க / உருவாக்க அனுமதி இல்லாவிட்டால், பிழைச் செய்தி காண்பிக்கப்படும்.

11.25 கோப்பை_எழுது

பயன்படுத்தும் விதம் ::= "கோப்பை_எழுது(" கோப்பு வகை மாறியின் பெயர் ", " மாறியின் பெயர் அல்லது மதிப்பு ")

இந்தக் கட்டளை ஏற்கெனவே திறக்கப்பட்ட ஒரு கோப்பில் எழுதப் பயன்படுகிறது. எழுத்துகள், எண்கள், தேதிகள் என்று பல விஷயங்களை இதன்மூலம் எழுதலாம். இவை அந்தக் கோப்பின் நிறைவுப் பகுதியில் சென்று சேரும்.

ஒருவேளை உங்களுக்கு அந்தக் கோப்பையில் எழுதுவதற்கான அனுமதி இல்லாவிட்டால், பிழைச் செய்தி காண்பிக்கப்படும்.

11.26 கோப்பை_மூடு

பயன்படுத்தும் விதம் ::= "கோப்பை_மூடு(" கோப்பு வகை மாறியின் பெயர் ")

இந்தக் கட்டளை திறந்த கோப்பை முற்றிலுமாக மூடப் பயன்படுகிறது. அதன்பிறகு அதை வாசிப்பதோ, அதில் எழுதுவதோ இயலாது. அப்படி ஒரு தேவை ஏற்பட்டால், மூடிய கோப்பை மீண்டும் இன்னொருமுறை திறப்பது அவசியம்.

தமிழில் நிரல் எழுது - எழில் நிரலாக்க மொழி

11.27 கோப்பை_படி

பயன்படுத்தும் விதம் ::= மாறியின் பெயர் “ = கோப்பை_படி(” கோப்பு
வகை மாறியின் பெயர் “)”

இந்தக் கட்டளை ஏற்கெனவே திறந்திருக்கும் ஒரு கோப்பில் உள்ள பகுதிகளை வாசிக்கப் பயன்படுகிறது. இப்படி வாசித்த வரிகள் அனைத்தும் ஒரு மாறியில் எழுத்துச் சரங்களாகச் சேமிக்கப்படும். அதன்பிறகு, அவற்றை அச்சிடலாம், அல்லது, அதே நிரலில் வேறு கணக்கீடுகளுக்குப் பயன்படுத்தலாம்.

```
*****
*** எழில் மொழி வழியாக ***
** நன்கு கணினி **
** நிரலாக்கம் கற்க **
*** *****
```