

அலகு - V

பகுப்பாய்வு வேதியியலில் - கணிப்பொறியின் பயன்பாடுகள்
(Computer Application in Analytical Chemistry)

முன்னுரை

கணிப்பொறி என்பது கணக்கீடுகளையும் (Calculations) கணிப்பீடுகளையும் (Computations) மிகவும் துல்லியமாகவும் வேகமாகவும், மின்னணுவின் (Electronics) கோட்பாடுகளை (Principles) பயன்படுத்திச் செயலாற்றும் ஒரு பொறியாகும்.

கணிப்பொறியின் வரலாறு

மனிதன் முதன் முதலாக வணிகத்தில் ஈடுபட்டபோது கணக்கீடு கருவியின் தேவையை உணர்ந்தான். இரண்டாம் உலகப்போரின் (1946) தொடக்கத்தில் பேபேஜ் வழங்கிய தத்துவத்தின் அடிப்படையில் John W. Mauchly மற்றும் Eckert என்பவர்களால் பென்கலவியா பல்கலைக்கழகத்தின் மூலம் (ENIAC) என்வழி மின் கணிப்பொறி வடிவமைக்கப்பட்டது. இதனைத் தொடர்ந்து கணக்கீடு கருவி பல பரிமாண வளர்ச்சியை அடைந்தது. பின்வரும் அட்டவணை கணிப்பொறியின் வளர்ச்சியினை விளக்குகிறது.

கணிப்பொறி வளர்ந்த முறை

வ.எண்	தலைமுறை	வருடம்	முன்னேற்றம்
1.		3500 BC	அபேகஸ் (மனிதனால் கணக்கீடும் கருவி)
2.		17ம் நூற்றாண்டின் தொடக்கத்தில் (1600-1623)	ஜான் நாபியரின் முதல் எந்திரக் கணிப்பான்
3.		1630	கலிலியோ என்ற அறிஞரின் நழுவுகோல் என்ற ஒரு கணப்பீட்டுக் கருவி
4.		1623-1662	பிளேய்ஸ் பாஸ்கல் கூட்டல், கழித்தல் செய்யவல்ல எந்திரக் கணிப்பான்
5.		1646-1716	வில்ஹெல்ம் லெபனீஸ் என்பவரின் கணப்பீட்டு கருவிகளின் குறைபாடுகளை நீக்கி கூட்டல், கழித்தல்,

6

1833

7.

1940-45

8.

முதல் தலைமுறை 1946-58

9.

இரண்டாம் தலைமுறை

1859-64

10.

மூன்றாம் தலைமுறை

1964-1971

11.

நான்காம் தலைமுறை

1972-1980

12.

ஐந்தாம் தலைமுறை

1980-1995

பெருக்கல், வகுத்தல் ஆகிய என்வழி இயக்கி தானியங்கு முறையில் இயங்கும் கணப்பீட்டுக் கருவி

என்வல் பெபேஜ்-ன் பகுப்பாய்வு பொறி கணக்கீட்டுக் கருவி

ஜோன்வாட் பார்க்கின் மின்னணு கணிப்பொறி கருவி.

ENIAC, EDVAC, EDSAC, SEAC மற்றும் IAS என்வழி பொறிக்க கருவி என்வழி மின்னணுக் கருவிகள் அமைக்கப்பட்ட கணிப்பொறிகள்.

TRADIC, IBM 1620 NCR 304, WHIRLWIND பொன்றவை திசுமையான பயன்படுத்தி உருவாக்கப்பட்ட கணிப்பொறி

சிறிய அளவு ஒருங்கிணைப்புத் திறமையு சிற்றுகளைக் கொண்டு அமைக்கப்பட்டன.

LSI, VLSI, மின்செய்லி ஆய்விற் கணிப்பொறி, தனிபாள் கணிப்பொறி பொன்றவை.

செயற்கை திறன்னைக் கொண்டு அமைக்கப்பட்ட CRAY XMP 415 மிக வேக கணிப்பொறி

கணிப்பொறி மற்றும் கணிப்பொறி நிரல்களுக்கான வரையறை :

கணிப்பொறி

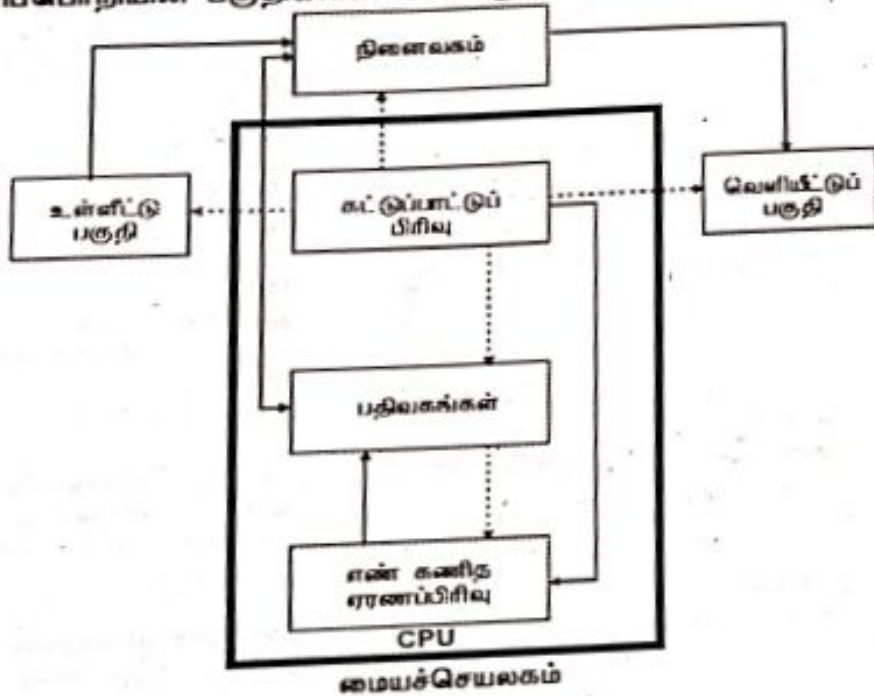
கணிப்பொறி என்பது தேக்கப்பகுதியில் தேக்கிவைக்கப்பட்டுள்ள நெறிமுறைகளின்படி கட்டளைகளைக் கொண்டு, கொடுக்கப்பட்ட நரவுகளைச் செயலாற்றி, வேண்டிய விளைவுகளை உண்டாக்கி, மனிதன் கைப்பட செய்வதைவிட வேகமாகவும் துல்லியமாகவும் செய்யவல்ல ஒரு மின்னணுக் கருவியாகும்.

கணிப்பொறி நிரல்கள் :

ஒரு குறிப்பிட்ட சிக்கலைத் தீர்க்க வேண்டுமெனில், நம்மால் நேரடியாக தயார் செய்து தரப்படும் வரிசைமுறை தகவலுக்கு கணிப்பொறி நிரல்கள் எனப்படும்.

BASIC, FORTRAN, PASCAL, C, C++, JAVA, COBOL, போன்ற உயர்நிலைமொழி வகையின் மூலமே தகவல்களை கொண்டு நிரல்கள் எழுதுகின்றனர். நம்மால் புரிந்துக் கொள்ளக்கூடிய அளவில், நிரல்களை பொறிமொழிக்கு மாற்றப்பட்டே கணிப்பொறி மூலம் செயல்படுத்தப்பட்டு வெளியீடுகள் பெறப்படுகின்றன.

கணிப்பொறியின் பகுதிகளை விளக்கும் தொகுதிப்படம்



..... தகவலின் பாய்வைக் குறிக்கின்றது
 ————— கவலின் கட்டுப்பாட்டுக் குறிப்புகளைக் குறிக்கின்றது

Basic units of a digital computer
 ஒரு கணிப்பொறியின் அடிப்படை பாகங்கள் பின்வருமாறு பகுதிகளைக் கொண்டதாகும்.

உள்ளீட்டுப் பகுதி

மையச் செயலகப் பகுதி

வெளியீட்டுப் பகுதி

உள்ளீட்டுப் பகுதி (Input Unit)

கட்டளைகளையும் டேட்டாக்களையும் கம்ப்யூட்டருக்குள் செலுத்தும் பண்பும் சாதனத்தையே இன்புட் யூனிட் என்கிறோம். கட்டளைகள் மற்றும் டேட்டாக்களை பைனி லடிவில் மாற்றி கம்ப்யூட்டரின் மெயினில் இன்புட் யூனிட் மூலம் செய்கிறது. பல வகையான இன்புட் யூனிட்கள் இருக்க வேண்டும். அவற்றுள் கோர்ட்டான் அதிக அளவில் பயன்படுத்தப்படுகிறது. இது உதாரணங்கள் : துப்புக் கேர்ட்டர் ரீடர் (OCR), மேக்னடிக் இயக் கேர்ட்டர் ரீடர் (MICR), மொஸ் (Mouse), விஷுவல் டிஸ்பிளே யூனிட் (VDU) போன்றவை.

மையச் செயலகப் பகுதி (CPU)

சிபியு-வை கம்ப்யூட்டரின் இருதயம் எனலாம். சிபியு-வின் முக்கிய பணி புரோகிராம்களை செயல்படுத்துவதே ஆகும். சென்ட்ரல் யூனிட்டில் (1) மெமரி யூனிட் (2) அரித்மெடிக் மற்றும் லாஜிக் யூனிட் (3) கன்ட்ரோல் யூனிட் ஆகியவை உள்ளன.

நினைவகம் (Memory Unit)

புரோகிராமையும் டேட்டாவையும் இங்கேதான் சேகரித்து வைக்க முடியும். மெமரி யூனிட் இறுதி முடிவுகளையும் அதற்கு முந்திய முடிவுகளையும் தற்காலிகமாக ஸ்டோர் செய்ய உதவுகிறது. எல்லா தகவல்களையும் புதிட்டில் குறியீடுகளாக மாற்றி தள்ளுகத்தே வைத்துக் கொள்ளும். டேட்டாவும் மற்ற தகவல்களும் மெயின் மெமரியில் இருந்து அரித்மெடிக் மற்றும் லாஜிக் யூனிட்டிற்கு (ALU) மாற்றப்படுகிறது. இவைகள் தேவை ஏற்படும்போது மெயின் மெமரியில் இருந்து ஆக்சிலரி மெமரிக்கும், ஆக்சிலரி மெமரியிலிருந்து மெயின் மெமரிக்கும் மாற்றப்படுகிறது.

சாதாரணமாகவே, மெயின் மெமரியின் அளவு குறைவாகவே இருக்கும். மேலும், மெயின் மெமரியில் சேகரித்து (ஸ்டோர்) வைக்கப்படும். தகவல்கள் எலக்ட்ரிக் பவர் ஆஃப் செய்யப்பட்டால் மறந்துவிடும்.

எண் கணித ஏரணப்பிரிவு (Arithmetic & Logic Unit - ALU)

இந்த சாதனம் கணக்கிடும் பணி, ஒப்பிடும் பணி மற்றும் லாஜிக்கல் பணிகளைச் செய்யப் பயன்படுகிறது. டேட்டாக்களை தேவை ஏற்படும்போது மெயின் மெமரியிலிருந்து எடுத்துக் கொள்ளும், கன்ட்ரோல் யூனிட்டில் இருந்து

வரும் சைகைகளுக்கு ஏற்ப எல்லா பணிகளையும் நிறைவேற்றி அவற்றின் முடிவுகளை மறுபடியும் மெயின் மெமரிக்கு அனுப்பும்.

கட்டுப்பாட்டு பகுதி (Control Unit)

கட்டுப்பாட்டு பகுதி கம்ப்யூட்டரின் இதர பாகங்களின் பணியை கண்காணிக்கிறது. மெயின் மெமரியில் உள்ள ஆணைகளை நிறைவேற்றும் பொறுப்பு கன்ட்ரோல் யூனிட்டைச் சார்ந்தது. கன்ட்ரோல் யூனிட் மெயின் மெமரியில் இருந்து வரும் ஆணைகளின் தன்மைக்கேற்றவாறு அவற்றை மாற்றி யின் நிறைவேற்றுகிறது. ஒவ்வொரு ஆணையையும் உணர்ந்து கொண்டு அதற்கேற்றவாறு மற்ற பகுதிகளுக்கு சைகைகளை அனுப்பி அவற்றை இயங்கச் செய்கிறது.

கட்டுப்பாட்டு பகுதியை நமது உடலின் மத்திய நரம்பு மண்டலத்துக்கு ஒப்பிடலாம். இதுவே மற்ற எல்லா பகுதிகளில் நடக்கும் பணிகளையும் கண்காணித்து ஒருங்கிணைக்கிறது. கட்டுப்பாட்டு பகுதியிலிருந்து கட்டுப்பாடு சைகைகளும், டைப்பிங் சைகைகளும் உருவாக்கப்பட்டு புரோகிராம நிறைவேற்றும் பொருட்டு இதர பகுதிகளுக்கு அனுப்பப்படுகிறது.

வெளியீட்டு பகுதி (Output Unit)

முடிவுகளை வெளி உலகுக்கு தெரிவிக்கும் யூனிட் அவுட்புட் யூனிட் எனப்படும். அவுட்புட் யூனிட் முடிவுகளை மெமரி யூனிட்டிலிருந்து பெற்றுக் கொண்டு அவைகளை நமக்கு தேவைப்படும் முறையில் கொடுக்கும். உதாரணங்கள் : விஷுவல் டிஸ்ப்ளே யூனிட், பிரிண்டர் பிளாட்டர்.

நெறிமுறைகளும் வரையில்களும்

தேர்வுபெற்ற ஆய்வாளர்களால், ஆரம்பகாலத்தில் கீழ்க்கண்ட கணிப்பொறி மொழி மூலமே கணிப்பொறி நிரல்கள் எழுதப்பட்டது. ஆனால் இன்றைய காலக்கட்டங்களில் கணிப்பொறியில் செயல்பாடுகளுக்கு பயன்படும் கணிப்பொறி நிரலை எழுத உயர்நிலை கணிப்பொறி மொழிகளே பயன்படுத்துகின்றனர். அவற்றில் சிலவகைகள் பின்வருவனவாகும்.

Fortran (formula translation) -1957

Fortran II

Fortran IV

Fortran 77

Fortran 90

Fortran 95

BASIC (Beginners all purpose symbolic instruction code)1967

Pascal

C (1970's)

C++ (1980's)

Java (1995)

C# (2000)

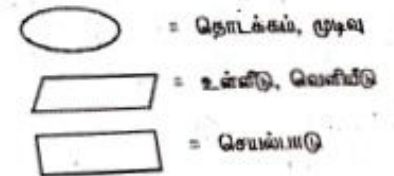
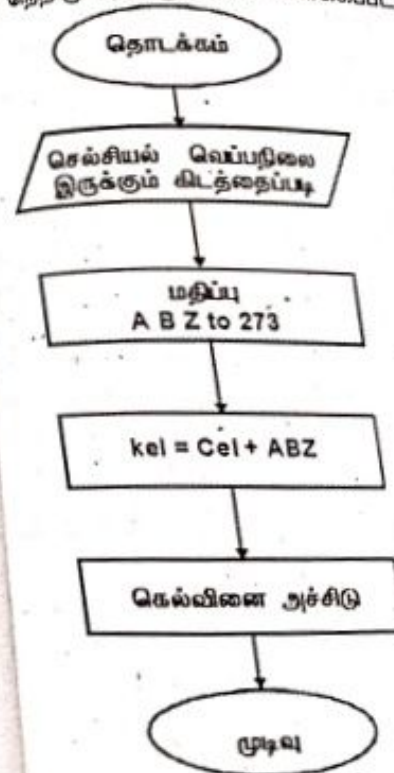
மேலும் சில மொழிகள் பயன்படுத்துகின்றனர்.

ஒவ்வொரு மொழியும் ஒரு தனிப்பட்ட தொடரியலும் ஒரே மாதிரியான நிரல்பயல்களையும் கொண்டுள்ளன. எல்லா மொழிகளிலும் வேறுபாடான நிரல்பயல்களையும் கொண்டுள்ளன. எல்லா மொழிகளிலும் வேறுபாடான நிரல்பயல்களையும் கொண்டுள்ளன. எல்லா மொழிகளிலும் வேறுபாடான நிரல்பயல்களையும் கொண்டுள்ளன. எல்லா மொழிகளிலும் வேறுபாடான நிரல்பயல்களையும் கொண்டுள்ளன.

BASIC VERSION (வழி):

```
10 REM CONVERSION OF CELSIUS TEMP. TO KELVIN TEMP.
20 INPUT "GIVE NAME OF PLACE AND TEMP. IN CELSIUS"; N$, CEL
30 ABZ = 273
40 KEL = CEL + ABZ
50 PRINT "THE TEMPERATURE OF"; N$; "IS="; KEL; "KELVIN"
60 END
```

இந்த நெறிமுறைகளுக்கான விளக்கப்படம்,



BASIC மொழியின் அறிமுகம்

Basic மொழி என்ற உயர்நிலை கணிப்பொறி மொழியை ஜான் ஜி. கெமினி மற்றும் தாமஸ் குரீட்ஸ் என்பவர்களால் 1967-ல் வடிவமைக்கப்பட்டது. அறிவியல் சார்ந்த துறைகளில் பயன்படும் மற்ற உயர்நிலைமொழிகள் FORTRAN, Pascal, Cand C++, Java, C# (C Sharp என்று அழைக்கப்படும்) முதலியனவாகும். அறிவியல் சார்ந்த வாய்ப்பாடுகளின் அளவுகளுக்கான மதிப்புகளை கணக்கிடுவதில் Basic, FORTRAN, C போன்ற உயர்நிலைமொழிகளும் ஒரு முதன்மையான பயன்பாடுகளாகும். Basic போன்ற நிரல்மொழிகளில் எழுத்துக்கள் (பேரெழுத்துகள் A to Z) மற்றும் சிறப்பு குறியீடுகளான +, -, /, *, ;, ^, %, & மற்றும் சில பயன்படுத்தப்படுகின்றன.

ஒவ்வொரு Basic கூற்றும் (வரிமொழி மாற்றிகளாக செயல்படுத்தல்) ஒரு கூற்று எண்ணையும், விளக்க குறிப்பு (Keyword) மற்றும் கட்டளைகளையும் கொண்டிருக்கும். ஒரு நிரலானது மாறிகளையும் (Variables) மாறிலிகளையும் (Constants) கொண்டிருக்கும். மாறிலிகளில் இருவகை உண்டு (எண்ணுருக்கள் மற்றும் எழுத்துச் சரங்கள்). அதுபோன்றே மாறிலிகளும் இரண்டு வகைப்படும்.

கணிதத்தில் நாம் அறிந்த சமன்பாடான $y = ax - b$ ல் y மற்றும் x ஆனது மாறிகளாகவும், a ஆனது மாறிலியாகவும் இருக்கும். ஒரு கணிப்பொறி நிரலில் உள்ளீடு கூற்றுகள், மதிப்பளி கூற்றுகள், கோவை மதிப்பீடு மற்றும் அச்சுக்கூற்றுகள் ஆகியவை காணப்படும்.

உயர்நிலை மொழிகள் (High Level Languages)

நாம் பேசும் மொழியைமப்பைக் கொண்ட, எளிதில் புரிந்துக் கொள்ளக் கூடிய நிரல்களை வடிவமைக்க உருவாக்கப்பட்ட கணிப்பொறி மொழிகள் உயர்நிலை மொழிகள் எனப்பட்டன. இவை கணிப்பொறி வகை சாரா மொழிகளாகும். அதாவது உயர் மொழியில் எழுதிய நிரல்களை, அந்த மொழியின் தொகுப்பி (Compiler) யுடன் எந்த வகைக் கணிப்பொறியில் வேண்டுமானாலும் செயலாக்கலாம். ஆனால் உயர்மொழியில் எழுதிய நிரல்களை கணிப்பொறி புரிந்து கொள்ள அந்த மொழித் தொகுப்பி (Compiler) அல்லது வரிமொழி மாற்றி (Interpreter) தேவை.

இப்பொழுது வெப்பநிலையை செல்சியஸ் வெப்பநிலையிலிருந்து கெல்வின் வெப்பநிலைக்கு மாற்றக்கூடிய ஒரு எளிய நிரலை காண்போம்.

C Version (வழி):

```
# Include <Stdio.h>
# Include <Math.h>
main ( )
```

```
{ charplace [20];
  int cel, kel;
  scanf ("% s%d", place, cel);
  abz = 273;
  kel = cel + abz;
  printf ("the kelvin temperature of % s is", place);
  printf ("=%d degree/n", kel);
}
```

இந்த நிரலில் KEL, ABZ, CEL ஆகியவை என்னுரு மாறிகளாகும். N\$ என்பது சர மாறியாகும். 273 என்பது ஒரு என்னுரு மாறியாகும். "THE TEMPERATURE OF", "IS =", "KELVIN", "GIVE NAME OF PLACE AND CHEMISTRY LABORATORY" ஆகியன சர மாறிலிகளாகும். N\$=PHYSICAL இயக்கும் பொழுது இதன் வெளியிடானது THE TEMPERATURE OF PHYSICAL CHEMISTRY LABORATORY IS298 KELVIN என்று அமையும்.

C மொழி :

'C' மொழி 1970-ம் ஆண்டு Bell Laboratories எனப்படும் அமெரிக்க கணிப்பொறி நிறுவனத்தில் பணிபுரிந்த டென்னிஸ் ரிட்சி என்பவரால் உருவமைக்கப்பட்டது முதலில் 'C' மொழி UNIX என்னும் பொறி இயக்க அமைப்பிற்கு உருவாக்கப்பட்டது அதன் பிறகு பிற இயக்க அமைப்புகளுக்கு உருவாக்கப்பட்டது மேலும் C மொழியில் சில விரும்பத்தக பண்புகள் உள்ளன. அது எவை என்பது கீழே வரிசைப்படுத்தப்பட்டுள்ளன.

- கேள்விக்குறி இயக்கி (Question mark operator)
- கண்ணி அமைப்பின் சுருக்கம் (Concise for - loop)
- மறுகழற்சி (Recursion)
- தடைக் கூற்று (Break)
- தொடர் கூற்று (Continue)
- குறிப்பீடு (Pointers)
- கட்டமைப்புகள் (Structures)
- தொகுப்புகள் (Unions)
- மேக்ரோ (Macros)

Turbo C கோப்புத் தொகுப்பிகளில் C நிரல் இயக்கப்பட பின்வரும் கோப்புகள் இணைக்கப்பட வேண்டும்.

```
# Include <Stdio.h>
# Include <Math.h>
```

C நிரலின் கட்டமைப்பானது கீழே காட்டப்பட்டுள்ளது.

C மொழியின் மிக முக்கியமான பண்புகளில் (?) கேள்விக்குறி இயக்கியும் ஒன்று. இது IF- THEN ELSE கூற்றுக்கு நிகரானது.

இந்த இயக்கியின் மாதிரி இயக்கப் படவம் :
Variable = (Condition) > exp 1 : exp 2 ;
(மாறி) (நிபந்தனை)

மேற்காணும் படவத்தில் நிபந்தனை உண்மையாகும் போது கோவை 1ம், நிபந்தனை பொய்யாகும் போது கோவை 2என இருக்கும். ஒத்த ஈரணு அல்லது வேறுபட்ட ஈரணு மூலக்கூறுகளில் நிறைக்குறைவை யின்வரும் திரல் மூலம் கணக்கிடுதல். $\mu(\text{homo})M_1/2N_0$ $\mu(\text{hetero}) = M_1 \cdot M_2 / ((M_1 + M_2) \cdot N_0)$ என்பது இவற்றின் வாய்ப்பாடாகும்.

கேள்விக்குறி இயக்கி (Question mark operator)
திரல் 1.

```
/* Program to calculate reduced mass of homo diatomic or hetero
diatomic. The number of */
/* inversion centers in the molecule is input. If number of inversion
centers is Zero */
/* the calculation is for hetero diatomic and if number of inversion centers
is one */
```

```
/*the calculation is for homo diatomic*/
#define AVO 6.023e23
```

```
main()
```

```
{
```

```
char molecule[20];
int nin;
double redm, m1, m2;
printf("enter name of molecule/n");
gets (molecule);
printf("enter no. of inv. centers and masses\n");
scanf("%d%if", &nin, &m1, &m2);
redm=(nin==0)?m1*m2/((AVO*(m1+m2)):m1(2.0*AVO);
printf("the reduced mass for %s is % 10.4e grams\n", molecule,
redm);
```

input and output
enter name of molecule

hydrogen

enter no. of inv. centers and masses

0 1.008 35.45

the reduced mass for hydrogen is 8.3679e-25 grams

enter name of molecule

hydrogen_chloride

enter no. of inv. centers and masses

0 1.008 35.45,

the reduced mass for hydrogen_chloride is 1.6273e-24 grams.

The advantage of C is that the 4 statements in FORTRAN or BASIC

30 IF NIN=0 THEN 40. ELSE 70

40 REDM=M1*M2/((AVO*(M1+M2))

50 GO TO 80

60 REDM = M1/(2*AVO)

80....

are replaced by the single statement

redm=(nin==0)?m1*m2/((AVO*(m1+m2)):m1/(2.0*AVO);

in C language.

மற்றொரு உதாரணமாக, இலட்சிய வாயுக்களின் வெப்பநிலை மாற நிலையில் வேலை செய்தல், வெப்பமாறா நிலையில் வேலை செய்தலில் கேள்வி குறி இயக்கி மூலம் செயல்படுத்த மேலும் ஒரு திரல் மற்றும் வெளிப்புரை கொடுப்பட்டுள்ளது.

கன்னி அமைப்பின் கருக்கம்
திரல் 2.

```
double awts[4]={39.0, 12.01, 14.008, 32.0};
main()
```

```
{
```

```
char molecule[20];
int i;
```

```
printf("enter name of molecule\n");
gets(molecule);
```

```
for (i=0;mw=0.0;i<3;mw+=awts[i],++i);
printf("the mol.wt.of %s is %7 3f grams/n", molecule, mw);
```

Input and Output

Enter name of molecule
KCNS

The mol.wt of KCNS is 97.018 grams

Unlike FORTRAN and BASIC the for loop in C can have multiple loop variables. The same program when written in BASIC requires 4 statements for the for loop as shown below:

```
30 MW=0.0
40 FORI=1 TO 4
50 MW=MW+AWT(I)
60 NEXTI
70....
```

பின்வரும் நிரலானது நான்கு அணு மூலக்கூறான ABCD (KCNS) என்ற நான்கு அணுக்களின் அணு எடையளவிலிருந்து மூலக்கூறு எடையை கணக்கிடப்படுகிறது.

GLOBAL மற்றும் LOCAL மாறிகள்

பின்வரும் நிரலானது Local மற்றும் Global மாறிகளை அறிமுகப்படுத்துவதோடு GLOBAL மாறியை பயன்படுத்தப்பட்டுள்ளது. C மொழியில் துணை செயல்முறையை நிரல்களில் ஒரேயொரு மாறி மட்டுமே Main திரலுக்கு திருப்பி அனுப்பப்படும்.

நாம் இதற்கு எடுத்துக்காட்டாக கெல்வின் வெப்பநிலையிலிருந்து செல்சியஸ் வெப்பநிலைக்கு

```
*main program*/
main()
{
```

```
int kelvin,ab_ze=273.celsius,Z;
printf("enter Kelvin temp.\n");
scanf("%d",& Celsius);
z=sub(kelvin, ab_ze);
printf("the Celsius temperature is %d degrees\n",z);
```

```
}
/*Function program*/
sub(x,y)
```

```
int x,y;
{
return(x-y);
}
```

இந்திரலில் மாறி ab-ze என்பது y-க்கு தகவெடுக்கப்படுகிறது. கெல்வின் மாறியின் மதிப்பு x-க்கு தகவெடுக்கப்படுகிறது. பிறகு செயல்நிலை $sub(x,y)$, x-y-ன் மதிப்பையே கணக்கிட்டு Main திரலுக்கு திருப்பியனுப்பிறது. இதில் return சுற்று ஒரேயொரு மதிப்பை மட்டுமே main திரலுக்கு திருப்பியனுப்பும் நாம் ஒரே சமயத்தில் இரண்டு சமன்பாடுகளைக் கொண்டு கணக்கிட இத்தகைய செயல்நிலை உபயோகப்படாது. அத்தகைய பயன்பாட்டிற்கு Global மாறிகளை நாம் பயன்படுத்த வேண்டியிருக்கும். பின்வரும் அச்சுவின்பல் அளவுருக்கள் E_a மற்றும் A ஆகியவற்றைக் கணக்கிட பயன்படுகிறது.

$$\text{சமன்பாடு } K = \exp(-E_a/RT)$$

இரண்டு வெப்பநிலைக்கு பின்வரும் சமன்பாடு எழுதப்படும்.

$$\ln K_1 = \ln A - E_a/RT_1$$

$$\ln K_2 = \ln A - E_a/RT_2$$

இந்த ஒத்த சமன்பாடுகளிலுள்ள தெரியாத E_a மற்றும் $\ln A$ னை தீர்க்க பின்வருமாறு நிரல் Global மாறிகளைக் கொண்டு அமைப்பு.

```
/*main program*/
#define R=8.314
```

```
double xx,yy,k1, k2, t1, t2;/*global variables*/
main()
{
```

```
char reaction[30];
double eact,pref;
printf("give the name of the reaction\n");
gets(reaction);
printf("give k1, k2, t1 and t2\n");
scanf("%f%f",&k1,&k2,&t1,&t2);
sim_eq();
pref=exp(xx);
eact=yy;
printf("the Arrhenius Parameters for %s are:\n", reaction);
printf("_____ \n");
printf("the energy of activation is % 10.4e joules \n", eact);
printf("the preexponential factor is %10.4e per sec\n", pref);
printf("_____ \n");
```

```
}
/*function program*/
```

```
/*x=ce-bf/(ae-bd); y=af-cd/(ae-bd);*/
sim_eq()/*function definition*/
```

```
{
    double a1, b1, c1, d1, e1, f1, z1;
    a1=1.0;
    b1=-1.0/(R*t1);
    c1=logk1;
    d1=1.0;
    e1=-1.0/(R*t2);
    f1=logk2;
    z1=a1*e1-e1-b1*d1;
    xx=(c1*e1-b1*f1)/z1;
    yy=(a1*f1-c1*d1)/z1;
}
```

Input and Output

Give the name of the reaction

Decomposition_of_benzaldehyde

Give k1, k2, t1 and t2

0.011 145.0 700.0 1000.0

the Arrhenius Parameters for Decomposition_of_benzaldehyde are

the energy of activation is 1.8403e5 joules

the preexponential factor is 5.9518e11 persec

மேற்கண்ட நிரலில் xx மற்றும் yy ஆகிய Global மாறிகள் (முதன்மை நிரல் மற்றும் செயல்நிரல் இரண்டிலும் பயன்படுகவை) பயன்படுத்தப்படுகின்றன. இந்த இரண்டு மாறிகளின் மதிப்புகளும் Global மாறிகளின் பயன்பாட்டைக் கொண்டு பரிமாற்றம் செய்யப்படுகின்றன. இதற்கு மற்றொரு வழியாக குறியீடுகள் (Pointers) கூற்றை கொண்டும் நிரல் எழுதலாம். அடுத்து வரும் உதாரண நிரலில் அர்ஜனியஸ் அளபுருக்களை பெற குறியீடுகளை பயன்படுத்தப்பட்டுள்ளது.

```
#define R=8.314
```

```
main()
```

```
char reaction [30];
```

```
double eact,pref, a,b,c,d,e,f,*a1, *b1, *c1, *d1, *e1, *f1;
```

```
/* *a1, *b1, *c1, *d1, *e1, and *f1 are pointers to the variables
```

```
,b,c,d,e and f*/
```

```
double k1, k2, t1, t2, x, y, & *x1, *y1, eact, pref;
```

```
a1 = *a; b1 = *b; c1 = *c; d1 = *d; e1 = *e; f1 = *f; x1 = *x; y1 = *y;
```

```
printf("give the name of the reaction/n");
```

```
gets (reaction);
```

```
printf("give k1, k2 t1 and t2/n");
```

```
scanf ("%f%f%f%f", &k1, &k2, &t1, &t2);
```

```
a = 1.0;
```

```
b = -1.0/(R*t1);
```

```
c = log k1;
```

```
d = 1.0;
```

```
e = -1.0/(R*t2)
```

```
f = logk2;
```

```
sim_eq(a1, b1, c1, d1, e1, f1, x1, y1);
```

```
pref = exp(xx);
```

```
eact = yy;
```

```
printf("the ArrheniusParameters for %s are :/n", reaction);
```

```
printf("_____/n");
```

```
printf("the energy of activation is % 10.4e joules/n", eact);
```

```
printf("the preexponential factor is % 10.4e per sec/n",pref);
```

```
Printf("\n\n");
```

```
/*function program*/
```

```
sim_eq(aa1, bb1, cc1, dd1, ee1, ff1, xx1, yy1)/* function definition*/
```

```
double *aa1, *bb1, *cc1, *dd1, *ee1, *ff1, *xx1, *yy1;
```

```
{
    double z1;
```

```
    a1 = 1.0;
```

```
    b1 = -1.0/(R*t1);
```

```
    c1 = log k1;
```

```
    d1 = 1.0;
```

```
    e1 = - 1.0 / (R*t2);
```

```
    f1 = logk2;
```

```
    z1 = (*aa1)*(*ee1) - (*bb1)*(*dd1);
```

```
    *xx1 = ((*cc1)*(*ee1) - (*bb1)*(*ff1)/z1;
```

```
    *yy1 = ((*aa1)*(*ff1) - (*cc1)*(*dd1)/z1;
```

```
}
```

Input and output

Give the name of the reaction

Decomposition_of_benzaldehyde

Give k1, k2, t1 and t2

0.011 145.0 700.0 1000.0

the ArrheniusParameters for Decomposition_of_benzaldehyde

the energy activation is 1.8403e5 joules

the preexponential factor is 5.9518e11 persec

C மொழிகள் மேலும் இரண்டு சிறப்பம்சங்களாக தடைக்கூற்று மற்றும் தொடர்கூற்றுக்களை குறிப்பிடலாம். தடைக் கூற்றானது ஒரு நிபந்தனைக்கு உட்பட்டு ஒரு கண்ணியின் செயல்பாட்டினை முடிப்பதற்கு பயன்படுகிறது. அதே வேலையில் தடைக் கூற்று வலையமைப்பு கண்ணியை உள்ளார்ந்த கண்ணியில் செயலை மட்டுமே கட்டுப்படுத்தும். தொடர்கூற்று ஒரு நிரலின் குறிப்பிட்ட படியமைப்பை மட்டுமே தாவுவதற்கு பயன்படுகிறது.

Input and Output

Enter name of nucleus

Hydrogen

Enter value of quadrupole moment

0.0

enter magnetic moment and spin values

2.79285 0.5

the NMR frequency for hydrogen is 42.571MHz

enter name of nucleus

Gallium_69

enter value of quadrupole moment

0.2318

the nucleus has quadrupole moment and control exits the loop

Continue கூற்று

ஒரு கண்ணியில் ஒரு பகுதியை விடுத்து கண்ணியின் அடுத்த கழற்சியைச் செயல்படுத்த Continue என்ற தொடர்கூற்றுப் பயன்படுத்தப்படுகிறது. தடை(break) கூற்று ஒரு கண்ணியில் கழற்சியை முழுக்க தடுத்துக் கட்டுப்பாட்டை கண்ணிக்கு வெளியே அனுப்பிலிடும். ஆகிறும் தொடர்கூற்று கண்ணியின் அடுத்த கழற்சிக்குக் கட்டுப்பாட்டை மாற்றுகிறது. தொடர்கூற்றைக் கொண்டு எழுதப்பட்ட நிரல் கீழே கொடுக்கப்பட்டுள்ளது.

நிரல் 7

```
/*The program uses continue statement*/
```

```
/*When this statement is used the specific iteration is skipped if condition is true and */
```

```
/*the control is transferred to carry out the next iteration of loop*/
```

```
/*This program counts the total number of fermions between atomic numbers 1 and 6*/
```

```
/*A fermion has odd number of nucleons and a Boson has even number of nucleons*/
```

```
main()
```

```
char atom[20];
int atnr, nentr, massnumber, count=0;
for(atnr=1; atnr<=6; ++atnr){
    printf("enter name of atom\n");
    gets(atom);
    printf("enter no. of neutrons\n");
    scanf("%d", &nentr);
    massnumber=atnr+nentr;
    if(massnumber%2==0){
        printf("the number of nucleons is even and");
        printf("nucleus is a Boson.control skips iteration\n");
        continue;
    }
    ++count;
    printf("the total number of nucleons between atomic numbers 1
and 6 is %d\n", count);
}
```

Input and Output

Enter name of atom

Hydrogen

Enter no. of neutrons

0

enter name of atom

Helium

Enter no. of neutrons

2

the number of nucleons is even and nucleus is a Boson. Control skips this iteration

enter name of atom

lithium

enter no. of neutrons

4

enter name of atom

beryllium

enter no. of neutrons

5

enter name of atom

boron

enter no. of neutrons
6

enter name of atom
carbon

enter no. of neutrons
6

the number of nucleons is even and nucleus is a Boson. Control skips this iteration

the total number of fermions between atomic numbers 1 and 6 is 4

மறு கழற்சி(Recursion)

மறு கழற்சி என்பது தனக்குத்தானே அழைத்துக் கொள்ளும் செயல்கூறுகள் ஆகும்.

C, C++, Pascal, Fortran go மற்றும் Java மொழிகளின் மிகமுக்கிய பயனாக மறு கழற்சியைக் குறிப்பிடலாம். மறுகழற்சியானது குறிப்பிட்ட கூற்றுகள்(Statements) தன்னையே அழைத்துக் கொள்ள பயன்படுகிறது. இவ்வகைக்கு உதாரணங்களாக எண்களின் தசம வரிசை கூட்டலையும், Factorial யையும் குறிப்பிடலாம்.

Factorial

$n! = n((n-1)!$ என்பது சமன்பாடாகும்.

உதாரணமாக, $4! = 4.3!$

/*Computing no. of valence diagrams for cyclic conjugated systems*/

/*using the formula $res = n! / ((n/2)! \times (n/2 + 1)!)$ */

main()

```
{
    char system[25];
    int res, npi, m, p, j;
    float fact ( );
    for (j=1; j<=3; ++j){
        printf("enter no. of pi electrons\n");
        scanf("%d", &npi);
        printf("enter name of system\n");
        gets(system);
        m=npi/2;
        p=m+1;
        res=fact(npi)/((fact(m)*fact(p)));
    }
```

```
printf("the number of valence diagrams for %s is %d\n", system,
res);
}
```

```
float fact (i)
int i;
```

```
{
    if(i==0)
        return(1);
    else
        return(i*fact(i-1));
}
```

Input and Output

Enter no. of pi electrons

6

enter name of system

benzene

the number of valence diagrams for benzene is 5

Enter no. of pi electrons

10

enter name of system

naphthalene

the number of valence diagrams for naphthalene is 42

Enter no. of pi electrons

14

enter name of system

anthracene

the number of valence diagrams for anthracene is 429

வேதியியலின் சில எளிய உதாரண நிரல்கள் :

1. a and b செல்சியஸிலிருந்து கெல்வின் வெப்பநிலைக்கு மாற்றதல் முறையே
Basic & C மொழிகளைக் கொண்டு செய்தல்.

$$Kel = CEL + 273$$

2) யீர் லாம்பர்ட் விதி

$$OD = \epsilon C l$$

3) அணு எடைகளிலிருந்து மூலக்கூறு எடைகள் காணல்-

$$M_{KCNS} = AWT_K + AWT_C + AWT_N + AWT_S$$

- 4) $Wiso = RT1n(V_2/V_1)$
 $Wadi = C_v(T_1 - T_2)$
 5) $res = n! / ((n/2)! \times ((n/2+1)!)$
 6)
 7) C, H, S, Cl, N மற்றும் O அணுக்களாகக் கொண்ட கரிமச்சேர்மங்களின் மூலக்கூறு எடை = $MWT = N1AWC + N2AWO + N3AWHH + N4AWCI + N5AWS + N6AWN$

இதில் N_1 முதல் N_6 வரை உள்ள அணுக்களின் எண்ணிக்கை AW.1 முதல் AW6 என்பன மூலக்கூறுகளின் அணு எடையையும் குறிக்கும்.

கருக்கம்

இந்த தொகுதியில் கணிப்பொறிப் பற்றிய அறிமுகம், தொகுதி விளக்க வரைபடமும், நிரல்கள் மற்றும் வரைவியல்கள், Basic மொழி நிரல்கள், C மொழியின் பண்புகள் மற்றும் Turbo C தொகுப்புகளைக் கொண்டு இயக்கப்பட்ட எடுத்துக்காட்டுடன் கூடிய நிரல்களும் கொடுக்கப்பட்டுள்ளன.

```
/* pgm to temperature conversion*/
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define ABZ 273.0
main()
{
    char place[20];
    float cel, kel;
    printf("\n\tEnter Name of Place: ");
    gets(place);
    printf("\n\n\tEnter CELSIUS Temperature: ");
    scanf("%f", &cel);
    kel = cel + ABZ;
    printf("\n\n\tThe Temperature for %s is %6.3f Degrees\n", place, kel);
    getch();
    return(0);
}
```

Input :
 Enter Name of Place : Chemistry Lab
 Enter CELSIUS Temperature: 27°C

Output :
 Temperature = 300 K

/* Pgm for BEER LAMBERT Law using C */

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
void main()
{
    char comp[20];
    double od, mab, len, con;
    clrscr();
    printf("\n\tEnter name of COMPLEX: ");
    gets(comp);
    printf("\n\tEnter values of od, mab and len: ");
    scanf("%lf%lf%lf", &od, &mab, &len);
    con = od / (mab * len);
    printf("\n\tThe concentration of %s is = %10.4e Molar", comp, con);
    getch();
}
```

Input:

Enter name of COMPLEX : $K_4[Fe(CN)_6]$
 Enter values of od, mab and len : 1, 7 & 2

Output:

The concentration of (complex) is = 4 Molar)

```
/* MolecularWeights from Atomic Weights */
/* M(KCNS) = AWT(k) + AWT(c) + AWT(n) + AWT(s) */
#include <stdio.h>
#include <conio.h>
double awt[5] = { 0.0, 39.0, 12.01, 14.008, 32.0 };
main()
```

```

char molecule[20];
int i;
double molwt;
clrscr();
printf("\nEnter Name of Molecule:");
gets(molecule);
for (i=1, molwt=0.0; i<5; molwt+=awt[i], i++);
printf("\nThe Molecular Weight of %s is = %6.3f Grams", molecule, molwt);
getch();
return(0);
}

```

Input :

Molecular Weights : 0.0, 39.0, 12.0, 14.008, 32.0

Output : 97.018 gm

/* Pgm to Demo Question Mark Operator in C */
/* Reversible expansion of a monoatomic GAS */

```

#include <stdio.h>
#include <math.h>
#include <conio.h>

```

```

#define R 8.314

```

```

void main()

```

```

{
char gas[20], type[20];
double t1, t2, cv, v1, v2, w;
clrscr();
printf("\n Enter Name of a Gas:");
gets(gas);
printf("\n Enter Type of Expansion:");
gets (type);
printf ("\n Enter values of t1, t2, v1, v2:");
scanf ("%lf%lf%lf%lf", &t1, &t2, &v1, &v2);
cv = 1.5 * R;
w = (t1==t2) ? R*t1*log(v2/v1) : cv*(t1-t2);
printf("\n The Work of %s for %s is=10.4e Joules", type, gas, w);
getch();
}

```

Input:

Enter Name of a Gas : Oxygen

Enter Type of Expansion : Isothermal

Enter values of t1, t2, v1, v2 : 273, 373, 2, 4

Output:

The Work of Isothermal expansion for oxygen is= 5.7632 Joules

/* RECURSION */

```

#include <stdio.h>

```

```

#include <math.h>

```

```

#include <conio.h>

```

```

main()

```

```

{
char system[20];
int res, m, p, n;
float fact();
clrscr();
printf("\n\n\tGive Name of System: ");
gets(system);
printf("\n\tEnter number of Pi Electrons:");
scanf("%d", &n);
m=n/2;
p=m+1;
res = fact(n) / (fact(m) * fact(p));
printf("\n\tThe No of Res Structures for %s is Equal to
%d", system, res);
getch();
return(0);
}

```

```

float fact(i)

```

```

int i;

```

```

{
if (i==0)
return(1);
else
return(i*fact(i-1));
}

```

Input :

Name of the System : Phenthrane

No. of Pi Electrons : 10

Output :

No. of resonance structures : 429

/* Linear Least Squares - fit method */

#include <stdio.h>

#include <math.h>

#include <conio.h>

#define R 8.314

double a[10],b[10];

main()

```

{
    char system[40];
    int n,n1,kk,i,n2;
    double t[10],k[10],c[10],sum1,sum2,sum3,sum4,sum5,sum6,d,ovd,
    std1,std2;
    double z2,z3,z4,z5,z6,z7,z8,z9,s1,s2,s3,s4,intercept,slope,pre_fac,enact;
    printf("enter no of systems for computation\n");
    scanf("%d",&n1);
    printf("linear least squares fit analysis for");
    printf("variation with temperature\n");
    for(i=1;i<=n1;++i)
        printf("enter name of reaction and no of data\n");
    for(i=1;i<=n1;++i)
    {
        printf("enter name of reaction and no of data\n");
        scanf("%lf",&t[i]);
        a[i]=1.0/t[i];
        printf("enter value of rate constant:");
        scanf("%lf",&t[i]);
        b[i]=log(k[i]);
    }
    clrscr();
    if(n1<=0)
    {
        for(i=1;i<=kk;++i)

```

```

        c[i]=1.0;
    }
    else
    {
        for(i=1;i<=kk;++i)
            c[i]=1.0/b[i];
    }
    sum1=0.0,sum2=0.0,sum3=0.0;
    sum4=0.0,sum5=0.0;
    for(i=1;i<=kk;++i)
    {
        z2=c[i]*a[i]*a[i];
        z3=c[i]*b[i];
        z4=c[i]*a[i]*b[i];
        z5=c[i]*a[i];
        sum1+=c[i];
        sum2+=z2;
        sum3+=z3;
        sum4+=z4;
        sum5+=z5;
        z6=sum2*sum3-sum5*sum4;
        z7=sum1*sum2-sum5*sum5;
    }
    intercept=z6/z7;
    z8=sum1*sum4-sum5*sum3;
    slope=z8/z7;
    sum6=0.0;
    for(i=1;i<=kk;++i)
    {
        z8=b[i]-intercept-slope*a[i];
        z9=z8*z8*c[i];
    }
    sum6+=z9;
    d=(double)kk;
    s1=sum6/(d-2.0);
    ovd=sqrt(s1);
    s3=s1*sum2/z7;
    std1=sqrt(s3);

```

Meer
20

```

s4=s1*sum1/z7;
std2=sqrt(s4);
pre_fac=exp(intercept);
enact=-slope*R;
printf("The preexponential factor and");
printf("energy of activation");
printf("for%s",system);
printf("are%10.4eM-1s-1and%10.4ejoules\n",pre_fac-enact);
printf("standard deviation of slope is %10.4e\n",std1);
printf("standard deviation of intercept");
printf("is%10.4e\n",std2);
printf("overall standards deviation is %10.4e/n",ovd);
getch();
}

```

/*Molecular weights of organic compounds having C,H,N,S,O and */

/*halogens*/

#include<stdio.h>

#include<math.h>

#include<conio.h>

main()

{

```

    char molecule[20];
    int i,n[6];
    float awt[6],molwt;
    printf("enter name of molecule\n");
    molwt=0.0;
    for(i=1;i<=6;++i)
    {
        printf("enter at.wt of ith element\n");
        scanf("%f",&awt[i]);
        printf("enter number of atoms of ith element\n");
        scanf("%d",&n[i]);
        molwt+=n[i]*awt[i];
    }
    clrscr();
    printf("the mol.wt. of %s is", molecule);
    printf("=%7.3f grams\n",molwt);
    getch();
    return(0);
}

```