அலகு I

பாடம் 1

செயற்கூறு

YXSVC8

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- செயற்கூறு வரையறை
- அளபுருக்கள் மற்றும் செயலுருபுக்கள்
- இடைமுகம் மற்றும் செயல்படுத்துதல்
- pure செயற்கூறுகள்
- பக்க விளைவுகள் ஆகியவற்றைப் பற்றி அறிந்து கொள்ளுதல்.

1.1 அறிமுகம்

ஒரு செயலைச் செய்து முடிப்பதற்கான நேரமே, நிரல் நெறிமுறையை எழுதுவதற்கும் மற்றும் மதிப்பீடு செய்வதற்கும் மிக முக்கியமான அடிப்படை ஆகும். நிரல் நெறிமுறையின் அர்த்தமுள்ள ஒப்பீட்டைப் பெறுவதற்கு (கணக்கீட்டு நேரத்தின்கால அளவானது) நிரலாக்க மொழி, நிரல் பெயர்ப்பி மற்றும் பயன்படுத்தும் கணினி ஆகியவற்றைச் சார்ந்திருக்கக்கூடாது. நிரலாக்க மொழியின் கூற்றுகளைப் பயன்படுத்தி நிரல் நெறிமுறைகள் வெளிப்படுத்தப்படுகின்றன என்பதை நீங்கள் அறிவீர்கள். மொத்தமான கூற்றுகள், பலமுறை மீண்டும் மீண்டும் செய்யப்பட வேண்டும் எனில், அந்த செயலைச் செய்து முடிப்பதற்காக துணை நிரல்கள் (Subroutines) பயன்படுகின்றன.

துணை நிரல்கள் கணினி மொழிகளின் அடிப்படைக் கட்டுமான தொகுதியாக விளங்குகின்றன. துணை நிரல்கள் என்பன ஒரு குறிப்பிட்ட செயலை மீண்டும் மீண்டும் செய்யப் பயன்படும் சிறிய நிரல் தொகுதியாகும். நிரலாக்க மொழிகளில் இத்துணை நிரல்கள் செயற்கூறுகள் (Functions) என்று அழைக்கப்படுகின்றன.

1.2 நிரலாக்க மொழியில் செயற்கூறுகள்

செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும். இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும். குறிப்பாக, குறிமுறையின் தொகுப்பைக் கொண்டிருக்கும். செயற்கூறானது பல வகை உள்ளீடுகளான மாறிகள் மற்றும் கோவைகளின் மீது செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.

1.2.1 செயற்கூறு வரையறை

a:=(24) என்ற உதாரணத்தைக் கவனிக்கவும். a:=(24) என்பது கோவையைக் கொண்டுள்ளது ஆனால் (24) என்பது கோவையல்ல. மாறாக, இது ஒரு செயற்கூறு வரையறை ஆகும். வரையறைகள், மதிப்புகளைப் பெயருடன் பிணைக்கின்றன. இங்கு, 24 என்ற மதிப்பு ‘a’ என்ற பெயருடன் பிணைக்கின்றது. வரையறைகள் கோவைகள் அல்ல, அதே நேரத்தில் கோவைகளை வரையறை எனக் கருதக் கூடாது. வரையறைகள் தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும். வரையறைகள் உள்ளமைவாக உள்ள கோவைகளைக் கொண்டதாகவோ அல்லது நேர்மாறாகவும் இருக்கலாம்.

1.2.2 அளபுருக்கள் மற்றும் செயலுருபுக்கள்

அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும்.

செயலுருபுக்கள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.

1.2.2.1 தரவு வகை இல்லா அளபுருக்கள்

செயற்கூறு வரையறைக்கு ஓர் எடுத்துக்காட்டைக் காணலாம்:

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

மேலேயுள்ள செயற்கூறு வரையறையில் 'b' என்ற மாறி அளபுரு ஆகும். மாறி 'b'க்கு அனுப்பப்படும் மதிப்பானது செயலுருபு ஆகும். செயற்கூறின் முன் நிபந்தனை (requires) மற்றும் பின் நிபந்தனை (return) கொடுக்கப்பட்டுள்ளது. நாம் எந்த தரவினத்தையும் குறிப்பிடவில்லை என்பதைக் நினைவில் கொள்க. சில மொழிகளின் நிரல் பெயர்ப்பி இவ்வகை சிக்கல்களை நிரல் நெறிமுறைப்படி சரி செய்கிறது. ஆனால் சில நிரல் பெயர்ப்பிக்கு தரவு வகையைக் குறிப்பிடுவது கட்டாயமாகும்.

மேலே உள்ள செயற்கூறு வரையறையில், if கோவை, then கிளைக்கு மதிப்பு 1 யைத் திருப்பி அனுப்பினால், தரவு வகை (data type) விதிப்படி if கோவை முழுவதுமே 'int' தரவு வகைக் கொண்டிருக்கும். if கோவையின் தரவு வகை int ஆக இருப்பதால் செயற்கூறின் திருப்பி அனுப்பும் மதிப்பும் int ஆக இருக்கும். 'b' யின் மதிப்பு சுழியத்தோடு = செயற்குறியுடன் ஒப்பீடு செய்யப்படுகிறது. அதனால் 'b' யின் தரவுவகையும 'int' ஆகும். * செயற்குறியுடன் 'a' யின் மதிப்பு மற்றொரு கோவையோடு பெருக்குத்தொகையைக் கணக்கிடுவதால் ,'a'-யின் வகையும் int ஆகும்.

1.2.2.2 தரவு வகையுடன் கூடிய அளபுரு

சில காரணங்களுக்காக, இப்பொழுது நாம் அதே செயற்கூறு வரையறை தரவு வகையுடன் எழுதலாம் :

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

'a' மற்றும் 'b' தரவு வகை குறிப்பு (type annotations) எழுதும் போது, அடைப்புக்குறிக்குள் ( ) அவசியமானது ஆகும். பொதுவாக, இந்த குறிப்புகளை நாம் விட்டுவிடலாம், ஏனெனில், நிரல்பெயர்ப்பி இவற்றை அனுமானிப்பது மிகவும் எளிது. முன்பெல்லாம் நாம் வெளிப்படையாகவே தரவு வகைகளை எழுதுவோம். எந்த வித அர்த்தமும் இல்லாத தரவு வகை பிழைச் செய்தியைப் பெறும் போது, இது மிகவும் பயனுள்ளதாகும். தரவு வகைக்கு வெளிப்படையாக தரவுவகை குறிப்பு எழுதுவது பிழைச் செய்தியைத் திருத்தம் செய்வதற்கு பயனுள்ளதாக இருக்கும். செயற்கூறை வரையறுப்பதற்கான தொடரியல், கணித பயன்பாட்டோடு நெருக்கம் கொண்டது. வரையறை let என்ற சிறப்புச் சொல்லோடு அறமுகப்படுத்தப்படுகிறது. அதனைத் தொடர்ந்து செயற்கூறின் பெயர் மற்றும் அதனுடைய செயலுருபுகள் பிறகு செயலுருபுவைக் கணக்கீடு செய்யும் வாய்ப்பாடு= குறிக்குப் பிறகு எழுத வேண்டும். தற்சுழற்சி செயற்கூற்றை வரையறுக்க விரும்பினால் let-க்குப் பதிலாக let rec என்று பயன்படுத்த வேண்டும்.

செயற்கூறு வரையறையின் தொடரியல்:

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

இங்கு 'fn' என்பது ஒரு மாறி ஆகும். இது செயற்கூறின் பெயரைக் குறிக்கும். குறிப்பெயராகும். 'a1' முதல் 'an' வரை உள்ள மாறிகள் அளபுருக்களைக் குறிக்கும் குறிப்பெயராகும். 'fn' என்பது தற்சுழற்சி செயற்கூறினால் rec என்ற சிறப்புச் சொல் தேவை, இல்லையெனில் அதை விட்டு விடலாம்.

குறிப்பு

தன்னைத் தானே அழைத்துக் கொள்ளும் செயற்கூறு வரையறைக்கு தற்சுழற்சி செயற்கூறு என்று பெயர்.

எடுத்துக்காட்டாக, உள்ளிடப்பட்ட எண் ஒற்றைப்படை எண்ணா அல்லது இரட்டைப்படை எண்ணா என சோதிக்கும் எடுத்துக்காட்டைக் காண்போம்.

```
(requires: x>= 0)
        let rec even x :=
        x=0 || odd (x-1)
        return 'even'
(requires: x>= 0)
        let odd x :=
        x<>0 && even (x-1)
        return 'odd'
```

செயற்கூறு வகைகளின் தொடரியல்

$x \rightarrow y$

$x1 \rightarrow x2 \rightarrow y$

$x1 \rightarrow \ldots \rightarrow xn \rightarrow y$

'x' மற்றும் 'y' மாறிகள் செயற்கூறு வகைகளைக் குறிக்கும். x → y என்ற செயற்கூறு வகையானவை 'x' வகையின் உள்ளீட்டைப் பெற்று 'y' வகையின் வெளியீட்டைத் திருப்பி அனுப்பும். அதே சமயம் x1 → x2 → y என்ற செயற்கூறு வகையானது இரண்டு உள்ளீட்டுகளைப் பெற்று, முதல் உள்ளீட்டின் வகை 'x1' ஆகும், இரண்டாவது உள்ளீட்டின் வகை 'x2' ஆகும். திருப்பி அனுப்பும் வெளியீட்டின் வகை 'y' ஆகும். x1 → … →xn → y யில் n அளபுருக்களின் உள்ளீட்டு வகை x ஆகும். வெளியீட்டின் வகை 'y' ஆகும்.

குறிப்பு

அனைத்து செயற்கூறுகளும் static வரையறையாகும். dynamic செயற்கூறு வரையறைகளே கிடையாது.

1.3 இடைமுகம் VS செயல்படுத்துதல்

ஒரு பொருள் (Object) செய்யக்கூடிய செயல்களின் தொகுப்பு இடைமுகம் ஆகும். எடுத்துக்காட்டாக, மின் விளக்கின் சுவிட்ச்சை அழுத்தும் போது மின் விளக்கு ஒளிர்கிறது. அது எவ்வாறு ஒளிர்கிறது என்பது கவலையில்லை. பொருள் நோக்கி நிரலாக்க மொழியில், இடைமுகம் என்பது அனைத்து செயற்கூறுகளின் விளக்கங்கள் (descriptions) ஆகும். ஒரு புதிய இடைமுகமாக இருப்பதற்கு இனக்குழு கண்டிப்பாக இவற்றைக் கொண்டிருக்க வேண்டும். நமது எடுத்துக்காட்டில், மின் விளக்கை போல் செயல்படும் எதுவும் turn_on() மற்றும் turn_off() என்ற செயற்கூறு வரையறையைக் கொண்டிருக்கும். X,Y,Z செயற்கூறுகளைக் கட்டாயமாக கொண்டிருக்கும் இனக்குழுவாகிய TYPE T (இடைமுகம் எதுவாக இருந்தாலும்) யின் பண்புகளை வலுக்கட்டாயமாக செயல்படுத்துவதே இடைமுகத்தின் நோக்கம் ஆகும்.

இனக்குழு அறிவிப்பானது வெளிப்புற இடைமுகத்தோடு அதன் உள்ளமை நிலை அந்த இடைமுகத்தைச் செயல்படுத்தும் செயல்பாட்டுடன் பண்புகளை உடைய குறிமுறை இணைக்கிறது.

பொருள் என்பது இனக்குழுவால் உருவாக்கப்பட்ட சான்றுரு ஆகும்.

வெளி உலகிற்கு பொருளின் காண்புநிலைப் பாங்கை இடைமுகம் வரையறுக்கிறது.

இடைமுகம் மற்றும் செயல்படுத்துதல் இரண்டுக்கும் இடையே உள்ள வேறுபாடு என்னவெனில்,

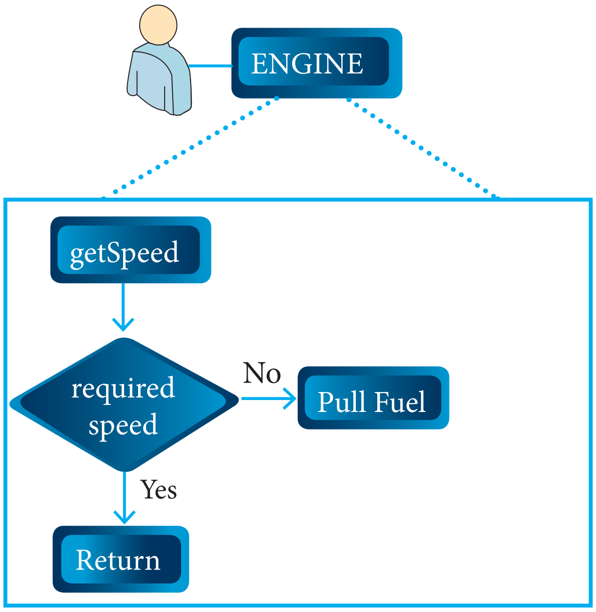
இடைமுகம்	செயல்படுத்துதல்
ஒரு பொருள் செய்யக்கூடிய நடவடிக்கையை வரையறுக்கிறது, ஆனால் அவற்றை உண்மையில் செய்யக் கூடியது இல்லை.	இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.

பொருள் நோக்கு நிரலாக்க மொழியில் இனக்குழு என்பது இடைமுகம் மற்றும் பொருள் எவ்வாறு செயல்படுத்தப்பட்டு நிறை வேற்றப்படுகிறது என்பதை செயல்படுத்துதல் ஆகும்.

1.3.1 இடைமுகத்தின் பண்புகள்

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

காரின் வேகத்தை அதிகரிக்கும் எடுத்துக்காட்டை எடுத்துக் கொள்வோம்.



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

உட்புறமாக, காரின் இயந்திரம் அனைத்து வேலைகளையும் செய்கிறது. எரிபொருள், காற்று, அழுத்தம் மற்றும் மின்சாரம் ஆகியவை இங்கு ஒன்றாக சேர்ந்து, ஆற்றலை உருவாக்கி, வாகனத்தை நகர்த்துகிறது. இந்த அனைத்து நடவடிக்கைகளும் கார் ஓட்டுநரிடம் இருந்து பிரித்து வைக்கப்பட்டுள்ளது. அவர் வேகமாக காரை செலுத்துவதில் ஆர்வமாக இருப்பார். இதனால், இடைமுகத்தை செயல்படுத்துதலில் இருந்து பிரித்து வைக்கப்படுகிறது.

ஒரு எளிமையான உதாரணத்தைக் காணலாம். கொடுக்கப்பட்டுள்ள 3 செயலுருபுகளில் குறைந்த மதிப்பைக் காணும் செயற்கூற்றைச் செயல்படுத்துவதைக் கவனி.

```
let min 3 x y z :=
        if x < y then
                if x < z then x else z
        else
                if y < z then y else z
```

1.4 Pure செயற்கூறுகள்

ஒரே மாதிரியான அளபுருக்களை அனுப்பும் போது, சரியான விடையைத் தரும் செயற்கூறு *pure* செயற்கூறுகள் ஆகும். எடுத்துக்காட்டாக, கணித செயற்கூறு *sin(0)*-ன் விடை எப்பொழுதுமே *0* ஆகும். இதன் அர்த்தம் என்னவென்றால், அதே அளபுருக்களைக் கொண்டு செயற்கூறினை ஒவ்வொரு முறையும் அழைக்கும் போது, அதே சரியான விடையை எப்பொழுதும் பெறலாம். மாறியின் பண்பை மாற்றக் கூடிய எந்த விதமான வெளிப்புற மாறியும் இல்லாமல் இருந்தால் அந்த செயற்கூறு *pure* செயற்கூறாகும்.

ஒரு எடுத்துக்காட்டை காண்க

```
let square x
        return: x * x
```

மேலேயுள்ள square செயற்கூறு pure செயற்கூறு ஆகும். ஏனென்றால் ஒரே மாதிரியான உள்ளீட்டிற்கு வேறுவித்தியாசமான வெளியீட்டைத் தராது.

கோட்பாடு சார்ந்த நன்மைகளை pure செயற்கூறுகள் கொண்டுள்ளன. இதன் ஒரு நன்மை என்னவென்றால், pure செயற்கூறாக இருக்கும் பொழுது, அதே அளபுருக்களுடன் செயற்கூறுவை பல தடவைகள் அழைக்கும்போது, உண்மையில் பெயர்ப்பிக்கு செயற்கூறுவை மீண்டும் ஒரு தடவை அழைக்கும் தேவை மட்டுமே ஏற்படுகிறது.

```
let i: = 0;
        if i <strlen (s) then
    -- Do something which doesn't affect s
        ++i
```

இது இயக்கப்படும் போது, ஒவ்வொரு முறையும் strlen (s) அழைக்கப்படுகிறது. strlen க்கு ஒட்டு மொத்தமாக ‘s’ தற்சுழற்ச்சி செய்ய தேவைப்படுகிறது. strlen என்பது pure செயற்கூறு என்றும், ‘s’யை மடக்கினுள் புதுப்பிக்காமலும் இருந்தால், strlen க்கு தேவைக்கு அதிகமான அழைப்பை நீக்கிவிட்டு, மடக்கை ஒரே ஒரு முறை செயல்படுததும். இதிலிருந்து நாம் தெரிந்து கொள்வது யாதெனில், strlen என்பது pure செயற்கூறாகும். ஏனென்றால், செயற்கூறு அளபுருவாக ஒரே ஒரு மாறியை எடுத்துக் கொண்டு அதனுடைய நீளத்தை கணக்கிடுகிறது. இந்த செயற்கூறு வெளி நினைவகத்தில் இருந்து உள்ளீட்டை எடுத்துக் கொள்கிறது. ஆனால் மதிப்புகளை மாற்றுவதில்லை திருப்பி அனுப்பும் மதிப்புகள் வெளி நினைவகத்தில் இருந்து பெறப்பட்டதாகும்.

குறிப்பு

pure செயற்கூற்றை மதிப்பீடு செய்யும் போது, எந்தவொரு பக்கவிளைவும் ஏற்படாது.

1.4.1 Impure செயற்கூறுகள்

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

```
let Random number
let a := random()
    if a > 10 then
        return: a
    else
        return: 10
```

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

1.4.2 பக்க விளைவுகள் (Impure செயற்கூறுகள்)

செயற்கூறு கவனிக்கத்தக்க வெளியுலக தொடர்பில் இருக்கும் போது பக்க விளைவுகள் ஏற்படும் என்பதை அறிவீர்கள். நம்முடைய நோக்கம் Pure செயற்கூறுவை உருவாக்குவதாக இருப்பினும் அது சில சமயங்களில் impure செயற்கூறுவாக மாறி விடுகிறது. பக்க விளைவு என்பது கேடு விளைவிக்கும் செயல் அல்ல, என்பதை நினைவில் கொள்க. சில சமயங்களில் அவை பயனுள்ளதாகும்.

செயற்கூறுவிற்கு வெளியில் மாறியை மாற்றுதல்

செயற்கூறுவின் வெளியே மாறியை மாற்றம் செய்வது என்பது பக்கவிளைவுகளில் ஒன்றாகும்.

எடுத்துக்காட்டு

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

மேலே கூறப்பட்ட எடுத்துக்காட்டில், y-ன் மதிப்பு செயற்கூறு வரையறையின் உள்ளே மாறுவதால் விடையானது ஒவ்வொரு முறையும் மாறும். inc() செயற்கூறுவின் பக்க விளைவு என்னவென்றால் வெளிப்புற மாறியான ‘y’ ன் மதிப்பை மாற்றுவதாகும். சில பக்க விளைவுகளை அடையாளம் காண்பது எளிதானதாகும் மற்றும் சில யோசிக்க தககுந்ததாக இருக்கும். நமது Impure செயற்கூறு எந்த அளபுருக்களையும் எடுத்துக் கொள்ளாது, எந்த மதிப்பையும் திருப்பி அனுப்பாது என்பது ஒரு நல்ல அறிகுறியாகும்.

கொடுக்கப்பட்ட அனைத்து எடுத்துக்காட்டுகள் மற்றும் விளக்கங்களில் இருந்து நாம் pure மற்றும் impure செயற்கூறுகளின் வேறுபாட்டை பின்வருமாறு காணலாம்.

Pure செயற்கூறு	**Impure செயற்கூறு**
Pure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்தே அமையும். அதனால், pure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் எப்பொழுதும் அதே திருப்பி அனுப்பும் மதிப்பே கிடைக்கும் இது எந்த பக்க விளைவுகளையும் கொண்டிருக்காது.	Impure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்து அமையாது. அதனால், impure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் வெவ்வேறான திருப்பி அனுப்பும் மதிப்பு கிடைக்கும்.
இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யாது.	இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யும்.

இரண்டு நேர்ம முழு எண்களின் மீப்பெரு வகுஎண்ணை கண்டுபிடிக்கும் pure செயற்கூறினை எடுத்துக்காட்டாக காணலாம்.

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

வெளியீடு
        gcd 13 27;
        - : int = 1
        gcd 20536 7826;
        - : int = 2
```

மேலே கொடுக்கப்பட்டுள்ள ‘gcd’ என்பது செயற்கூறுவின் பெயர் ஆகும். மாறி ‘b’ ன் மதிப்பு 0 ஆகும் வரை செயற்கூறு தன்னைத் தானே அழைத்துக் கொள்ளும் b மற்றும் (a mod b) ஆகிய இரண்டு அளபுருக்களை gcd செயற்கூறுவினுள் ‘a’ மற்றும் ‘b’ க்கு அனுப்புவதை நினைவில் கொள்க.

14.3 செயற்கூறுவை பயன்படுத்தி குரோமிலேண்டில் பச்சோந்திகள் என்ற சிக்கல் தீர்த்தல்

பதினொன்றாம் வகுப்பில் படித்த குரோமிலெண்டில் பச்சோந்திகள் என்ற பகுதியை நினைவில் கூர்க. இரண்டு வகையான பச்சோந்திகள் சமமான எண்ணாக இருந்தால், இந்த இரண்டு வகையும் சேர்ந்து மூன்றாவது வகையின் நிறத்தை மாற்றுவதற்கான நிரல் நெறிமுறையை உருவாக்கவும். முடிவில் அனைத்தும் ஒரே நிறத்தை காண்பிக்க வேண்டும். ஒவ்வொரு வகை பச்சோந்தியின் எண்ணிக்கையையும் a,b,c மாறிகளாக எடுத்துக் கொள்வோம். அதனுடைய தொடக்க மதிப்பு முறையே A, B மற்றும் C . a = b என்பது உள்ளீட்டு பண்பு ஆகும். இதன் உள்ளீட்டு வெளியீட்டுக்கான தொடர்பு a = b = 0 மற்றும் c = A+B+C ஆகும். இதனுடைய நிரல் நெறிமுறைக்கு monochromatize எனப் பெயரிடலாம். இதை monochromatize (a , b , c) என குறிப்பிடலாம்.

-- உள்ளீடுகள்	:	a = A, b = B, c = C, a = b
-- வெளியீடுகள்	:	a = b = 0, c = A+B+C

ஒவ்வொரு சுழற்சி நிலையிலும் 2 வகையான ஒரே எண்ணைக் கொண்ட பச்சோந்திகள் சந்தித்து மூன்றாவது வகைக்கு அதன் வண்ணத்தை மாற்றும். எடுத்துக்காட்டாக, A, B, C = 4, 4, 6, எனில் சந்திப்பின் வரிசையானது பின்வருமாறு,

iteration	a	b	c
0	4	4	6
1	3	3	8
2	2	2	10
3	1	1	12
4	0	0	14

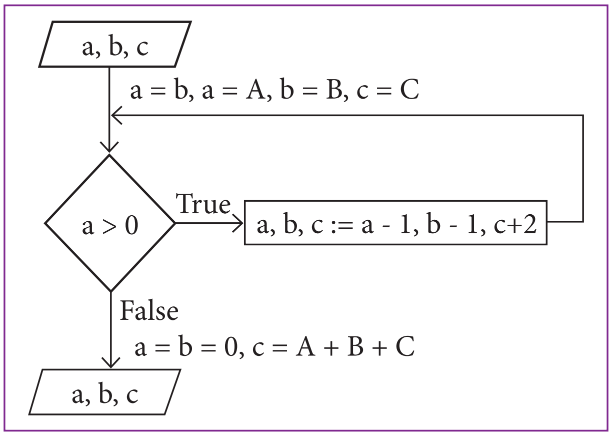
ஒவ்வொரு சந்திப்பிலும், a மற்றும் b-ன் மதிப்பு ஒன்று குறைகிறது மற்றும் c-ன் மதிப்பு 2அதிகரிக்கிறது. இதற்கான தீர்வு சுழற்சி நெறிமுறையில் வெளிப்படுத்துகிறது.

monochromatize (a, b, c)

-- உள்ளீடுகள்	:	a = A, b = B, c = C, a = b
-- வெளியீடுகள்	:	a = b = 0, c = A+B+C

```
while a > 0
a, b, c := a -1, b -1, c+2
```

இந்த நெறிமுறை பாய்வுப்படமாக கீழே காண்பிக்கப்பட்டுள்ளது.



இப்பொழுது நிரல் நெறிமுறையை செயற்கூறுவைப் பயன்படுத்தி எழுதலாம்.

```
let rec monochromatize a  b  c :=
        if a > 0 then
a, b, c := a-1, b-1, c+2
        else
           a:=0, b:=0, c:= a + b + c
        return c
```

நினைவில் கொள்க

- நெறிமுறைகள் நிரலாக்க மொழியின் கூற்றுகளைப் பயன்படுத்தி வெளிப்படுத்தப்படுகின்றன.
- ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக மீண்டும் மீண்டும் பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே துணை நிரலாக்கமாகும்.
- செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும். இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும்.
- செயற்கூறானது பல வகை உள்ளீடுகளைக் கொண்டு செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.
- வரையறைகள் தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும்.
- அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும். செயலுருபுக்கள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.
- தரவு வகைக்கு குறிப்பு எழுதும் போது, செயற்கூறு வரையறையில் () அடைப்புக்குறிகள் அவசியமானதாகும்.
- ஒரு பொருள் செய்யக்கூடிய செயல்பாடுகளின் தொகுப்பு இடைமுகம் ஆகும். இது நடவடிக்கையை வரையறுக்கிறது. ஆனால், உண்மையில் அவற்றை செய்யக்கூடியது இல்லை.
- செயல்படுத்துதல் என்பது இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.
- ஒரே மாதிரியான அதே அளபுருக்களை அனுப்பும் போது, சரியான விடையைத் தரும் செயற்கூறு pure செயற்கூறு எனப்படும்.
- செயற்கூறுக்கு அளபுருக்களை அனுப்பாத போதும், செயற்கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும். இந்த வகைச் செயற்கூறு impure செயற்கூறு எனப்படும்.

செய்முறைப் பயிற்சி

1. மூன்று எண்களில் சிறிய எண்ணை கண்டுபிடிப்பதற்கான செயற்கூறு வரையரையை கொண்ட நெறிமுறையை எழுதுக.
2. n வரையுள்ள எண்களின் கூட்டுத்தொகையைக் கணக்கிடும் தற்சுழற்சி செயற்கூறு வரையறையைக் கொண்ட நெறிமுறையை எழுதுக.

DJ995R

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே

(அ) துணை நிரல்கள் (ஆ) கோப்புகள்

(இ) Pseudo குறிமுறை (ஈ) தொகுதிகள்

2. பின்வரும் எந்த அலகு ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்பட்டுள்ளது?

(அ) துணை நிரல்கள் (ஆ) செயற்கூறு

(இ) கோப்புகள் (ஈ) தொகுதிகள்

3. பின்வரும் எது தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும்?

(அ) துணை நிரல்கள் (ஆ) செயற்கூறு

(இ) வரையறை (ஈ) தொகுதிகள்

4. செயற்கூறு வரையறையில் உள்ள மாறிகள் எவ்வாறு அழைக்கப்படுகிறது?

(அ) துணை நிரல்கள் (ஆ) செயற்கூறு

(இ) அளப்புருக்கள் (ஈ) செயலுருபு

5. செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் எவ்வாறு அழைக்கப்படுகிறது?

(அ) செயலுருபுகள் (ஆ) துணை நிரல்கள்

(இ) செயற்கூறு (ஈ) செயற்கூறு

6. தரவு வகை குறிப்பு எழுதும்போது, எது கட்டாயமாகிறது?

(அ) { } (ஆ) ( )

(இ) [ ] (ஈ) < >

7. பின்வரும் எது ஒரு பொருள் செய்ய வேண்டியதை தீர்மானிக்கிறது?

(அ) இயக்க அமைப்பு (ஆ) நிரல் பெயர்ப்பி

(இ) இடைமுகம் (ஈ) தொகுப்பான்

8. பின்வரும் எது இடைமுகத்தில் வரையறுக்கப்பட்ட கட்டளைகளை நிறைவேற்றுகிறது?

(அ) இயக்க அமைப்பு (ஆ) நிரல் பெயர்ப்பி

(இ) செயல்படுத்துதல் (ஈ) தொகுப்பான்

9. ஒரே மாதிரியான அதே அளபுருக்களை செயற்கூறுவிற்கு அனுப்பினால் சரியான விடையைத் தரும் செயற்கூறு எவ்வாறு அழைக்கப்படும்?

(அ) Impure செயற்கூறு (ஆ) Partial செயற்கூறு

(இ) Dynamic செயற்கூறு (ஈ) Pure செயற்கூறு

10. அளபுருக்களை அனுப்பும் போது பக்க விளைவுகளை ஏற்படுத்தும் செயற்கூறு எவ்வாறு அழைக்கப்படும்?

(அ) impure செயற்கூறு (ஆ) Partial செயற்கூறு

(இ) Dynamic செயற்கூறு (ஈ) Pure செயற்கூறு

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. துணைநிரல் என்றால் என்ன?
2. நிரலாக்க மொழியைப் பொறுத்து செயற்கூறுவை வரையறுக்கவும்.
3. X:=(78) இதன் மூலம் அறிவது என்ன?
4. இடைமுகத்தையும், செயல்படுத்துதலையும் வேறுபடுத்துக.
5. பின்வருவனவற்றுள் எது சாதாரண செயற்கூறு வரையறை மற்றும் எது தற்சுழற்சி செயற்கூறு வரையறை

```
i) let rec sum x y:
        return x + y
ii) let disp :
        print 'welcome'
iii) let rec sum num:
        if (num!=0) then return num + sum (num-1)
        else
        return num
```

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. இடைமுகத்தின் பண்புக்கூறுகள் யாவை?
2. strlen ஏன் pure செயற்கூறு என்று அழைக்கப்படுகிறது?
3. impure செயற்கூறுவின் பக்க விளைவுகள் யாவை? எடுத்துக்காட்டுடன் விளக்குக?
4. pure மற்றும் impure செயற்கூற்றை வேறுபடுத்துக.
5. ஒரு செயற்கூறிக்கு வெளியே ஒரு மாறியை மாற்றினால் என்ன விளைவுகள் ஏற்படும்? ஒரு எடுத்துக்காட்டு தருக.

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. செயலுருபுகள் என்றால் என்ன?

   (அ) தரவுவகை இல்லாத அளபுருக்கள்

   (ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் விவரி?

2. பின்வரும் நிரலில்

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

   அ) செயற்கூறுவின் பெயர்

   ஆ) தற்சுழற்சி செயற்கூறு கூற்று

   இ) அளபுருக்கள் கொண்ட மாறியின் பெயர்

   ஈ) செயற்கூறுவை தற்சுழற்சிக்கு அழைக்கும் கூற்று

   உ) தற்சுழற்சியை முடிவுக்கு கொண்டுவரும் கூற்று ஆகியவற்றை எழுதுக.

3. pure மற்றும் impure செயற்கூறுவை எடுத்துக்காட்டுடன் விளக்குக.

4. இடைமுகம் மற்றும் செயல்படுத்துதலை எடுத்துக்காட்டுடன் விளக்குக.

REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, RobertoTamassia and Michael H. Goldwasser.*
2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*
3. *https://www.python.org*