ghl«-1 bra‰TW

2 kÂ¥bg© édh¡fŸ

1) Ji z ãuš v‹whš v‹d?

Ji z ãušfŸ v‹gJ F¿Kiwæ‹ xUmyF MF«. ÏJ xU F¿¥Ã£l braiy gšntW ãiyfëš Û©L« Û©L« brŒa ga‹gL« Á¿a ãuš bjhFÂahF«. ÏJ bra‰TWfŸ, KiwikfŸ v‹W miH¡f¥gL«.

2) ãuyh¡f bkhêæš bra‰TW tiuaiw v‹whš v‹d?

bra‰T‰¿‹ tiuaiwfŸ v‹gJ gy nfhitfis cŸsikthf bfh©lit, Ïit kÂ¥ò¡fis bgaUl‹ mjhtJ kh¿Íl‹ Ãiz¡»‹wd. Ïªj tiuaiwfŸ jd¡bfd xU jåahd bghJ tot¤Âid (bjhlçaš) bfh©lit.v.fh

a := (24) š

24 v‹w kÂ¥ò a v‹w bgaUl‹ Ãiz¡f¥gL»wJ.

3) X:=(78) Ïj‹ _y« m¿tJ v‹d?

X:=(78) v‹gJ xU bra‰TW tiuaiwahF«. Ï§F 78 v‹w kÂ¥ò X v‹w bgaUl‹ Ãiz¡f¥gL»wJ.

4) ÏilKf«, brašgL¤Jjš ntWgL¤Jf

ÏilKf« - xU bghUis brŒa¡Toa elto¡iffis tiuaW¡»wJ Mdhš brašfis brašgL¤jhJ. v.fh Ïd¡FG

ga‹gL¤Jjš – ÏilKf¤Âš tiuaW¡f¥g£LŸs brašfis brašgL¤J»‹wJ (v.fh: Ïd¡FGé‹ _y« cUth¡f¥gL« bghUŸfŸ)

5) Ã‹tUtd t‰¿š vJ rhjhu z bra‰TW k‰W« j‰RH‰¢Á bra‰TWvd f©l¿f

1. let rec sum x y :
   return x+y
   - Ï§F rec v‹w Kj‹ik brhšYl‹ m¿é¡f¥g£L ÏU¥ÃD« ÏÂš j‹id jhnd miH¤J¡bfhŸS« T‰W« cŸsJ vdnt ÏJ rhjhu z bra‰TW tiuaiw
2. let disp :
   print ' welcome '
   - ÏJ rhjhu z bra‰TW tiuaiw
3. let rec sum num :
   if (num!=0) then return mum + sum(num-1)
   else return num
   - Ï§F rec v‹w Kj‹ik brhšYl‹ m¿é¡f¥g£L nky« j‹id jhnd miH¤J¡bfhŸS« sum(num-1) T‰W« cŸsJ vdnt j‰ z RH‰¢Á bra‰TW tiuaiw

_‹W kÂ¥bg© édh¡fŸ

1. ÏilKf¤Â‹ g©ò TWfŸ ahit?

- Ïd¡FG v‹w th®¤ÔU ÏilKf«, xU bghUis Kiwahf cUth¡fÎ« mjid brašgL¤jÎ« njita ahd thŒ¥òfis tH§F»wJ
- bghUŸfë‹ cW¥ò bra‰TWfis ga‹gL¤Â, bghUŸfë‹ g©òfisÍ«, g©ò TWfisÍ« f£LgL¤j cjÎ»wJ

2. strlen bra‰TW V‹ pure bra‰TW vd miH¡f¥gL»wJ?

- Xnu khÂçahd braYUòfis mD¥ò« nghJ xnu éilia ÂU¥Ã jU« bra‰TW pure bra‰TW MF«.
- Strlen() v‹w bra‰TW bfhL¡f¥gL« ru¤Â‹ Ús¤ij f z ¡»l ga‹gL tjhš Ïj‰F bfhL¥ggL« braYUÃš kh‰w« Ïšyhj tiu xnu kÂ¥ig ÂU¥Ã jU« vdnt ÏJ pureš bra‰TW v‹W miH¡f¥gL»wJ.

3. Impure bra‰T‰¿‹ g¡f éisÎfŸ ahit ? vL¤J fh£Ll‹ és¡Ff

Impure bra‰T‰¿‹ ÂU¥Ã mD¥ò« kÂ¥ò msòU¡fis k£L« bghW¤J mikahJ. mÂš ga‹gL¤jg£LŸs KGjGé bt ë¥òw kh¿fŸ k‰W« bra‰TWfis bfh©L ÂU¥Ã mD¥ò« kÂ¥ò x›nthU Kiw« khW«. v.fh

www.Padasalai.Org

1.random() v‹w bra‰TW x›bthU Kiwmi H¡f¥gL« khWg£l j‹njh‹ ¿a jdkhd v© f is ÂU¥Ã mD¥ò«.

2. y =1

```
let inc x:=
    y: = x + y
    return y
```

- v‹w Impure bra‰TW btëkh¿ahd y ia mo¥gilahf bfh©L Ïa§Ftjhš x›bthU Kiw miH¡F« nghJ« bt›ntW kÂ¥òfis ÂU¥ò«

4. Pure k‰W« Impure bra‰TWfis ntWgL¤Jf.

Pure bra‰TW	Impure bra‰TW
Pure bra‰TW ÂU¥Ã mD¥ò« kÂ¥ò K‰¿Y« msòU¡fis bghW¤J mikÍ«.	Impure bra‰TWé‹ ÂU¥Ã mD¥ò« kÂ¥ò K‰¿Y« msòU¡fis bghW¤J mikahJ.
bra‰T¿id xnu braYUòfis¡ bfh©L miH¤jhš v¥bghGJ« xnu kÂ¥Ãid ÂU¥Ã mD¥ò«	bra‰T¿id xnu braYUò¡fis¡ bfh©L miH¤jhY« bt›ntwhd kÂ¥ig ÂU¥Ã mD¥ò«.
kl¡F T‰Wfëš Ïjid braYUòfis òÂ¥Ã¡fhkš ga‹gL¤Âdhš x›bthU KiwÍ« miH¥gij jé®¤J giHa kÂ¥igna ãušbga®Ã ga‹gL¤Â bfhŸS«	v¤jifa NHèY« x›bthUKiwÍ« miH¡f¥g£L »il¡F« òÂa kÂ¥òfis ãušbga®Ã ga‹gL¤Â bfhŸS«
ÏJ g¡féisÎfis V‰gL¤¤jhJ	ÏJ g¡féisÎfis V‰gL¤J«.

5. XU bra‰TW¡F btëna xUkh¿ia kh‰¿dhš v‹d éisÎfŸ V‰gL«? vL¤J¡fh£L jUf.

XU bra‰TWé‹ btëòw kh¿ia kh‰w« brŒtjhš V‰gL« g¡féisÎ v‹gJ mJ ÂU¥Ã mD¥ò« éilahdJ btëòw kh¿æ‹ g¡féisédhš xnu kÂ¥ig bfh©L miH¤jhY« x›bthU KiwÍ« khW«.

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

- Ï§F y- v‹gJ btë¥òw kh¿ahF« ÏJ bra‰T‰¿š ga‹gL¤j¥g£LŸsJ. Ïªj btë¥òw kh¿ x›bthUKiw miH¡f¥gL« nghJ« jdJ kÂ¥ig kh‰¿¡bfhŸtjhš inc()bra‰T¿‹ ÂU¥Ã mD¥ò« éilahdJ btëòw kh¿æ‹ g¡féisédhš xnu kÂ¥ig bfh©L miH¤jhY« x›bthU KiwÍ«khW«.

IªJ kÂ¥bg© édh éilfŸ

1. msòU¡fŸ k‰W« braYUò¡fŸ v‹whš v‹d?

m) juÎ tiff Ïšyhj msòU¡fŸ

M) juÎ tiffÍl‹ Toa msòU¡fis g‰¿ vGJf

msòU¡fŸ : bra‰TW tiuaiwæš cŸs kh¿fŸ.

braYUòfŸ : bra‰TW tiuaiwæš cŸs kh¿èfS¡F mD¥g¥gL« kÂ¥ò¡fshF«.

m) juÎ tiff Ïšyhj msòU¡fŸ

bra‰T‰¿š kh¿fë‹ juÎ tif F¿¥Ãl gl éšiy v‹whY« Áy ãuš bga®¥ÃfŸ , bra‰T¿š ga‹gL¤j¥g£LŸs T‰WfŸ, kh¿èfŸ, k‰W« nfhitfis bfh©L juÎ tifia ãuš be¿Kiwæ‹ go Ô®khå¤J¡bfhŸ»‹wd

v.fh.

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

- Ï§F bv‹gJ msòU thF« b¡F mD¥g¥gL« kÂ¥ò braYÒ.

nk‰f©l vL¤J¡fh£oš b¡F juéd« F¿¥Ãlééšiy, ÏU¥ÃD« bra‰TWt iuaiwæš cŸs

- if nfhit, then » is 1 v‹w ‘int” kÂ¥ig ÂU¥Ã mD¥òtjhš, juÎ tif éÂ¥go (data type) nfhit KGtJnk ‘int’ juÎ tifahF«.
- ’b’ ‹ kÂ¥ò b=0 vdRêa¤njhL x¥Õl¥gLtjhš, ’b’ ‹ kÂ¥ò« ‘int’ juÎ tifahF«.
- ’a’ ‹ kÂ¥ò int tifahd ’b’ Îl‹ bgU¡f¥gLtjhš, a æ‹ tifÍ« int MF«.

M) juÎ tifÍl‹ Toa msòU¡fŸ

bra‰T‰¿š kh¿fë‹ juÎ tif btë¥gilahf F¿¥Ãl¥ gL« nghJ jutif rh®ªj jU¡f ÃiHfŸ k‰W« F¿KiwÃiHfŸ j逌f¥gL» ‹wJ.

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

- ‘a’ k‰W« ‘b’ juÎ tifæid¡ F¿¥ÃL« nghJ ( )mil¥ò¡F¿¡FŸ F¿¥gJ mtÁakhF«.

2) Ã‹tU« ãuèš

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

m) br‰TWé‹ bga®
M) j‰RH‰Ábra‰TW
Ï) msòU¡fŸbfh©lkh¿æ‹ bga®
<) bra‰TWit j‰RH‰Á¡Fmi H¡F« T‰W
c) j‰RH‰ÁiaKoé¡Fbfh©LtU« T‰W
M»at‰¿id vGJf.

m) br‰TWé‹ bga® :- gcd
M) j‰RH‰Á bra‰TW :- let rec gcd a b :=
Ï) msòU¡fŸ bga® :- a, b
<) bra‰TWit j‰RH‰Á¡F miH¡F« T‰W :- gcd b (a mod b)
c) j‰RH‰Áia Koé¡F bfh©L tU« T‰W :- if b <> 0 ‹ else return a

3. Pure k‰W« Impurebra‰TWfis vL¤J¡fh£Ll‹ étç.

1. Pure bra‰TWfŸ.

Xnu khÂçahd msòU¡fis mD¥ò« nghJ xnu éilia jU« bra‰TW Pure bra‰TWfshF«. ÏJ bra‰TW khh¿ mšyJ tiuaiwæ‹ g©ig kh‰w¡ Toa vªj éjkhd btë¥òw kh¿Í« Ïšyhkš ÏUªjhš mªj bra‰TW Pure bra‰TWfshF«.

v.fh 1.

- sin (0) - ‹ éilv¥nghJnk 0.
- sin (90) - ‹ éilv¥nghJnk 1 jU«

v.fh 2.

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

- Ï¢bra‰TW Pure bra‰TWfshF«. Vbd‹whš xnu khÂçd cŸÇ£L¡F ntW é¤Âahrkhd btëp£oid¤ juhJ.

2. Impure bra‰TWfŸ

bra‰TW¡F msòU¡fis mD¥ò« nghJ, bra‰T¿‹ cŸns cŸs kh¿ahdJ g¡féisÎfis (btsp£oš kh‰w«) V‰gL¤Âdhš Ïªjbra‰Tiw Impure bra‰TWfŸvdyh«. Ïªj tif bra‰TW x›bthU Kiw miH¡F« bghGJ« bt›ntW éjkhd btëp£oid¡ bfhL¡F«.

v.fh 1.

random() v‹gJ Impure bra‰TW x›bthU Kiwͫ miH¡F« nghJ v‹déil bfhLF« vd ã¢rakhf Tw ÏayhJ.

v.fh 2.

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

Ï§F y- v‹gJ btë¥òw kh¿ahF« ÏJ bra‰T‰¿š ga‹gL¤j¥g£LŸsJ. Ïªj btë¥òw kh¿ x›bthUKiw miH¡f¥gL« nghJ« jdJ kÂ¥ig kh‰¿¡bfhŸtjhš inc()bra‰T¿‹ ÂU¥Ã mD¥ò« éilahdJ btëòw kh¿æ‹ gj¡féisédhš xnu kÂ¥ig bfh©L miH¤jhY« x›bthU Kiwͫ khW«.

4. Ïil Kf¤ij Í«, bra‰gL¤Jjiy Í« vL¤J¡fh£Ll‹ és¡F f.

Ïil Kf« (Interface)

ஒரு பொருள் செய்யக்கூடிய செயல்பாடுகளின் தொகுப்பு இடைமுகம் ஆகும். இது பொருளின் செயல்பாடுகளை வரையறுகிறது. ஆனால் அதனை நேடியாக செயல்பாடுத்தாது. எ.கா இனக்குழு

- வெளிஉலகிற்கு, பொருள்களின் காண்பு நிலைப் பாங்கை இடைமுகம் வரையறுக்கிறது
- இனக்குழுவின் பண்புகளை வலுக்கட்டாயமாக செயல்படுத்தவதே இடைமுகத்தின் நோக்கம் அகும்
- நிரலாக்க மொழியில் இடைமுகம் என்பது செயற்கூறுகளின் விளக்கங்களை குறிக்கிறது

பயன்படுத்துதல் (Implementation)

Ïil Kf¤Âš tiuaW¡f¥g£LŸs bra‰fis bra‰gL¤J»‹wJ.

(v.fh: bghUŸfŸ Ïd¡FGé‹ _y« cUth¡f¥g£l rh‹W)

எடுத்துக்காட்டு:

- ஒரு On / Off சுவிட்ச் என்பது இடைமுகம் எனக்கொண்டால் இது turn_on(), turn_off() என்ற செயற்கூறு வரையறைகளை கொண்டு இருக்கும். அதாவது விளக்கு எரிவதற்கான மின் சுற்றுகளை கொண்டுள்ளது. ஆனால் On/off என்ற செயல்பாட்டை மட்டும் வெளி உலகிற்கு காட்டுகிறது இருப்பினும் அது தானே On / Off செயல்பாடுகளை செய்ய இயலாது.
- சுட்விட்சின் On/Off செயற்கூறுகளை வெளியிலிருந்து இயக்கப்படும் போது அது விளக்கு எரிதல் அல்லது நிறுத்துதல் செயலை செய்கிறது இது **பயன்படுத்துதல்** ஆகும்.

பிற வினாக்கள்

1. bra‰TWfŸ அல்லது துணைநிரலின் j‹ikfŸ

1. bra‰TW v‹gJ F¿Kiwæ‹ xUmyF MF«.
2. ÏJ bgU«ghY« xU bgça F¿Kiw f£lik¥Ãš tiuaW¡f¥gL«.
3. bra‰TWfŸ gytif cŸÇLfshd kh¿fŸ k‰W« nfhitfë‹ ÛJ bra‰g£L ãiyahd btëp£il¤ jU»wJ.
4. முறையாக தொகுத்து வைத்துக்ககொண்டால் பலவேறு இடங்களில் மீண்டும் மீண்டும் பயன்படுத்தி கொள்ளலாம்

2. தற்சுழற்சி செயற்கூறு என்றால் என்ன?

தன்னைத்தானே அழைத்துக்கொள்ளும் செயற்கூறு வரையறைக்கு தற்சுழற்ச்சி செயற்கூறு என்று பெயர்

எ.கா

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

3. **செயற்கூறு வரையறையின் தொடரியலை எழுதுக.**

bra‰TWt iuaiwj fhdbjhlçaš

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

இங்கு

let - செயற்கூறு வரையறை bjhl¡f«.

rec - தற்சுழற்ச்சி செயற்கூறின் அடையாளம் காட்டும் விருப்ப பகுதி

fn - bra‰T¿‹ bga®

a1,a2,..an - msòU¡fŸ.

K - = ¡F ÃwF braYÙit¡ fz¡»L« thŒ¥ghLfis bfh©l bra‰TW tiuia vGjnt©L«.

எ.கா

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

bra‰TW bjhlçaš tiffŸ

$x \rightarrow y$

$x_1 \rightarrow x_2 \rightarrow y$

$x_1 \rightarrow \ldots \rightarrow x_n \rightarrow y$

x → உள்ளீட்டு அளபுரு வகை

y→ வெளியீட்டு தரவு வகை

4. **மூன்று எண்களில் பெரிய எண்ணை கண்டறியும் செயற்கூறு வரையறை எழுதுக**

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

5. KGv© fë‹ Û¥bgU tFv© iz (gcd) fhQ jš.

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

6. Pure bra‰TWfë‹ e‹ikfŸ / braš gL« éj«

- kl¡Ffëš Pure bra‰TWfis ga‹gL¤j¥gL« nghJ braYWòfŸ òJ¥Ã¡fhkš ÏUªjhš x›nthU Kiwí« bra‰TWfis miH¥gij jé®¤J, bra‰TW¡F flªj Kiw ga‹gL¤j g£l giHa cŸÇLfS¡F Û©L« mjid miH¡fhkš giHa btp£ila ga‹gL¤jÂ¡bfhSŸ«.
- nji tæšyhj Å© miH¥òfis Fiw¡»wJ
- ãuè‹ ntf¤ij mÂfgL¤J»wJ

ஒரு வரி குறிப்புகள்

- செயலை செய்யும் செய்து முடிக்கும் நேரத்தில் முக்கியமான நேரம் -
- o நெறிமுறை எழுதுவதற்கும், மதிப்பீடு செய்வதற்கும் ஆகும் நேரம்
- செயற்கூறு என்பது குறிமுறையின் மேலும் பிரிக்க முடியாத ஒரு சிறிய செயலை குறிக்கும் அலகாகும்
- அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள்
- செயலுருபுகள் எனப்து அளபுருக்கு அனுப்பபடும் மதிப்புகள்
- தரவு வகை குறிப்பிட படுதல் தருக்க பிழை மற்றும் கணக்கீட்டு பிழைகளை திருத்த பயன்படுகிறது
- மறு சுழற்ச்சி செயற்கூறுகளுக்கு rec என்ற சிறப்பு சொல் பயன்படுத்தபடுகிறது
- செயற்கூறுகள் அனைத்தும் நிலையான (Static) வரையறைகளை கொண்டவை.
- இனக்குழு என்பது செயற்கூறு வரையறைகளை கொண்ண வார்ப்புரு
- பொருள் என்பது இனக்குழுவால் உருவாக்கப்பட்ட சான்றுரு