

ជំពូកទី ១

សេចក្តីផ្តើមនៃភាសាសរសេរកម្មវិធី C

១- ប្រវត្តិនៃភាសា C

- ភាសាសរសេរកម្មវិធី C គឺជាភាសាសរសេរកម្មវិធីបែប structure មួយបង្កើតឡើង នៅទីពិសោធន៍ Bell ក្នុងឆ្នាំ ១៩៧២ ដោយលោក Dennis Ritchie។
- លក្ខណៈពិសេស របស់ ភាសាសរសេរកម្មវិធី C មានប្រភពមកពីភាសាមុនហៅថា B (Basic Combined Programming Language -BCPL)។
- ភាសាសរសេរកម្មវិធី C ត្រូវបានបង្កើតឡើងសម្រាប់ប្រើលើប្រព័ន្ធប្រតិបត្តិការ UNIX។
- នៅឆ្នាំ ១៩៧៨ Dennis Ritchie និង Brian Kernighan បានបោះពុម្ពផ្សាយលើកទី១ មានចំណងជើងថា "ភាសាសរសេរកម្មវិធី C" និងជាទូទៅត្រូវបានគេស្គាល់ឈ្មោះអ្នកនិពន្ធថា K&R C។
- នៅឆ្នាំ ១៩៨៣ វិទ្យាស្ថាន American National Standards Institute (ANSI) បានរៀបចំគណៈកម្មការមួយ ដើម្បីផ្តល់នូវការកំណត់អត្ថន័យដ៏ទូលំទូលាយនិងទាន់សម័យរបស់ C ។ លទ្ធផលទទួលបានពីអត្ថន័យ គឺ ស្តង់ដារ ANSI ឬ "ANSI C" ដែលត្រូវបានបញ្ចប់ទាំងស្រុងនៅឆ្នាំ ១៩៨៨។

២- លក្ខណៈពិសេសរបស់ភាសាសរសេរកម្មវិធី C

ភាសាសរសេរកម្មវិធី C មានអានុភាពលើផ្នែក៖

- Reliability : ភាពគួរឱ្យជឿជាក់បាន
- Portability : ភាពងាយស្រួលក្នុងការយកចេញពីកន្លែងមួយទៅប្រើនៅកន្លែងមួយទៀតបាន
- Flexibility : ភាពដែលអាចប្រែប្រួលតាមកាលៈទេសៈបាន
- Interactivity : ភាពមានអំពើឬសកម្មភាពទៅវិញទៅមកបាន
- Modularity : ភាពអាចបំបែកជាផ្នែកតូចៗបាន
- Efficiency and Effectiveness : ការឱ្យលទ្ធផលល្អនិងមានប្រសិទ្ធភាពខ្ពស់

៣- អត្ថប្រយោជន៍នៃភាសាសរសេរកម្មវិធី C

ភាសាសរសេរកម្មវិធី C ត្រូវបានគេប្រើសម្រាប់បង្កើតកម្មវិធីឱ្យប្រព័ន្ធកុំព្យូទ័រដែលបង្កបានជាចំណែកសំខាន់ៗរបស់ប្រព័ន្ធប្រតិបត្តិការ ដូចជា Windows, UNIX និង Linux។ ឧទាហរណ៍៖

- Database systems (ប្រព័ន្ធគ្រប់គ្រងទិន្នន័យ)
- Graphics packages (សរសេរបណ្តុំក្រាហ្វិក)
- Word processors (កម្មវិធីវាយអត្ថបទ)
- Spreadsheets (កម្មវិធីវាយទិន្នន័យមានលក្ខណៈជាតារាង)
- Operating system development (សម្រាប់អភិវឌ្ឍន៍ប្រព័ន្ធប្រតិបត្តិការកុំព្យូទ័រ)
- Compilers and Assemblers (សម្រាប់សរសេរ compilers និង assemblers កុំព្យូទ័រ)
- Network drivers (សម្រាប់ drivers បណ្តាញកុំព្យូទ័រ)
- Interpreters (សម្រាប់សរសេរ interpreters កុំព្យូទ័រ)

៤- កម្រិតបំណាច់នៃភាសាសរសេរកម្មវិធីកុំព្យូទ័រ

តើភាសាសរសេរកម្មវិធី C ត្រូវបានចាត់ទុកជាភាសាក្នុងកម្រិតណា?

គេចែកភាសាសរសេរកម្មវិធីជាបីកម្រិត គឺ៖

- ភាសាកម្រិតកណ្តាល (middle level languages) ៖ វាពុំបានផ្តល់នូវ functions ដែលមានស្រាប់ ដូចមានក្នុងភាសាកម្រិតខ្ពស់ឡើយ ប៉ុន្តែវាផ្តល់នូវ blocks ដែលអាចបង្កើតអ្វីៗទាំងអស់តាមតម្រូវការរបស់យើង ដើម្បីឲ្យបានលទ្ធផលដែលយើងចង់បាន។ ឧទាហរណ៍៖ ភាសា C, C++។
- ភាសាកម្រិតខ្ពស់ (high level languages) ៖ វាផ្តល់នូវអ្វីៗស្ទើរតែទាំងអស់ដែលអ្នកសរសេរកម្មវិធីត្រូវការប្រើ ដូចមានរួចជាស្រេចក្នុងភាសា។ ឧទាហរណ៍៖ Java, Python។
- ភាសាកម្រិតទាប (low level languages) ៖ វាគ្មានផ្តល់អ្វីសោះ ក្រៅពីចូលប្រើម៉ាស៊ីនដោយប្រើសំណុំ instruction សំខាន់ៗ។ ឧទាហរណ៍៖ Assembly language។

៥- លំនាំនៃភាសាសរសេរកម្មវិធីកុំព្យូទ័រ

ភាសាសរសេរកម្មវិធី C ជាភាសាមានលំនាំបែបជា Structured។ គេចែកភាសាសរសេរកម្មវិធីជាបីលំនាំធំៗគឺ លំនាំ structure oriented language, លំនាំ object-oriented language និងលំនាំ non-structure oriented language។

- លំនាំ structure oriented language ៖ ភាសាលំនាំប្រភេទនេះគឺ កម្មវិធីធំៗត្រូវចែកជាកម្មវិធីតូចៗ ហៅថា functions។ Prime Focus គឺស្ថិតនៅ function និង procedure ដែលប្រតិបត្តិលើទិន្នន័យ។

ទិន្នន័យផ្លាស់ទីជុំវិញប្រព័ន្ធដោយសេរីពី function មួយទៅ function មួយទៀត។ ទម្រង់កម្មវិធីមួយ អនុវត្តតាមវិធីពីលើចុះក្រោម។ ឧទាហរណ៍៖ ភាសា C, ភាសា Pascal, ភាសា ALGOL និង Modula-2។

- លំនាំ object-oriented language ៖ ភាសាប្រភេទនេះ កម្មវិធីត្រូវបំបែកទៅជា object។ Prime Focus គឺស្ថិតនៅទិន្នន័យ (data) ដែលត្រូវប្រតិបត្តិ ហើយមិនប្រតិបត្តិនៅ functions និង procedures ឡើយ។ ទិន្នន័យត្រូវបានបិទបាំង ហើយមិនអាចចូលប្រើបានដោយ functions ខាងក្រៅទេ។ ទម្រង់កម្មវិធីមួយ អនុវត្តតាមវិធី ពីក្រោមឡើងលើវិញ។ ឧទាហរណ៍៖ ភាសា C++, ភាសា Java, និង ភាសា C#។
- លំនាំ non-structure oriented language ៖ វាពុំមានទម្រង់ជាក់លាក់សម្រាប់សរសេរកម្មវិធីចំពោះភាសានេះ។ ឧទាហរណ៍៖ BASIC, COBOL, FORTRAN ជាដើម។

៦- របៀបសរសេរកម្មវិធីតាមភាសា C

នៅក្នុងផ្នែកខាងក្រោមបង្ហាញពីការសរសេរកម្មវិធីកុំព្យូទ័រសាមញ្ញមួយជាភាសា C ដើម្បីពន្យល់ពីការទទួលបានលទ្ធផល “Hello, world” ហើយនិងរបៀបត្រូវធ្វើមុនសរសេរកម្មវិធីមួយ។

- មូលដ្ឋានគ្រឹះនៃពាក្យបញ្ជារបស់ភាសា C សម្រាប់សរសេរកម្មវិធី C មួយ
- គំរូកម្មវិធី C តូចមួយដោយមានលទ្ធផលនិងការពន្យល់
- ជំហានសំខាន់សម្រាប់សរសេរកម្មវិធី C និងដើម្បីទទួលបានលទ្ធផល
- ការបង្កើត, ការធ្វើ Compilation និងការប្រតិបត្តិនៃកម្មវិធី C មួយ
 - របៀបដំឡើង C compiler និង IDE សម្រាប់ដំណើរការ codes របស់ភាសា C
- ទម្រង់សំខាន់ៗរបស់កម្មវិធី C
 - ឧទាហរណ៍កម្មវិធី C ដើម្បីធ្វើការប្រៀបធៀបគ្រប់ផ្នែកទាំងអស់
 - អធិប្បាយផ្នែកនិមួយៗរបស់កម្មវិធី C មួយ

៦.១- មូលដ្ឋានគ្រឹះនៃភាសា C សម្រាប់សរសេរកម្មវិធីមួយ

កម្មវិធីខាងក្រោមបានប្រើ syntax មួយចំនួនរបស់ភាសាសរសេរកម្មវិធី C ។

```

#include <stdio.h>
#include <conio.h>
void main() {
    /* some comments */
    printf("Hello World! ");
    getch();
}

```

ពាក្យបញ្ជាសំខាន់ៗរបស់ភាសា C	សេចក្តីពន្យល់
#include <stdio.h> #include <conio.h>	នេះជាពាក្យបញ្ជា preprocessor ដែលបញ្ចូលនូវ header file (stdio.h) សម្រាប់ input output មានលក្ខណៈស្តង់ដារ ពី C Library មុនធ្វើការ Compile កម្មវិធី C។ header file (conio.h) សម្រាប់ console input output ពេលគឺប្រើ getch()
int main()	នេះជាអនុគមន៍ main ដែលជាចំណុចចាប់ផ្តើមនៃកម្មវិធី C
{	សញ្ញានេះបញ្ជាក់ប្រាប់ថា ការចាប់ផ្តើមនៃអនុគមន៍ main
/* some comments */	អ្វីក៏ដោយដែលបានសរសេរខាងក្នុងពាក្យបញ្ជា "/* */" ក្នុងកម្មវិធី C ណាមួយនោះ នឹងមិនត្រូវគិតចំពោះ
printf("Hello World! ");	ពាក្យបញ្ជា printf បោះបង្ហាញលទ្ធផលមកលើអេក្រង់
getch();	ពាក្យបញ្ជានេះរង់ចាំការបញ្ចូលតួអក្សរណាមួយពី Keyboard
return 0;	ពាក្យបញ្ជានេះបញ្ចប់កម្មវិធី C (main function) និង ឲ្យ លទ្ធផល 0។
}	សញ្ញានេះបញ្ជាក់ប្រាប់ថា ទីបញ្ចប់ផ្តើមនៃអនុគមន៍ main

៦.២- កម្មវិធី C តូចមួយ

កម្មវិធី C ខាងក្រោមនេះមានលក្ខណៈងាយស្រួលជាមូលដ្ឋានគ្រឹះនៃភាសាសរសេរកម្មវិធី C។ កម្មវិធីនេះបង្ហាញលទ្ធផល "Hello World!" មកលើអេក្រង់។ គ្រប់ syntax និងពាក្យបញ្ជានៅក្នុងភាសា

សរសេរកម្មវិធី C មានលក្ខណៈប្រកាន់តួអក្សរតូចធំ។ ឃ្លានិមួយៗត្រូវតែបញ្ចប់ដោយសញ្ញា ; ដែលជាសញ្ញាបញ្ចប់ឃ្លាមួយ។

```
#include <stdio.h>

void main() {
    /* Our first simple C basic program */
    printf("Hello World! ");
    getch();
}
```

៦.៣- ជំហានសំខាន់ៗសម្រាប់សរសេរកម្មវិធី C និងសំនេរដល់ការបង្ហាញលទ្ធផល

ជំហានសំខាន់ៗខាងក្រោមនេះជាលក្ខណៈទូទៅរបស់ភាសាសរសេរកម្មវិធី C មិនថាកម្មវិធីមានលក្ខណៈតូចឬធំឡើយ ត្រូវអនុវត្តតាម ដើម្បីឈានទៅដល់ការបង្ហាញលទ្ធផល។

- Create (បង្កើតកម្មវិធី)
- Compile (ធ្វើការ compile)
- Execute or Run (ធ្វើការប្រតិបត្តិ ឬ ដំណើរការកម្មវិធី)
- Get the Output (ទទួលបានលទ្ធផល)

៦.៣.១- អ្វីទៅការសរសេរកម្មវិធី?

ការសរសេរកម្មវិធី C មួយដោយប្រើពាក្យនិងសញ្ញាផ្សេងៗដែលមនុស្សអាចយល់បាន។ គេអាចវាយបញ្ចូលកម្មវិធីទៅក្នុងកម្មវិធី editor ណាមួយ ហើយរក្សាឯកសារនេះទុកដោយដាក់ឈ្មោះ file មួយនិង extension .c សម្គាល់ភាសាកម្មវិធី C។ ជួនកាល គេដាក់ extension .cpp ប្រែប្រួលទៅតាម compiler ដែលយើងប្រើ។ file នេះត្រូវបានហៅថា source file។

៦.៣.២- អ្វីទៅការធ្វើ Compiling នៃកម្មវិធី C មួយ?

គេមិនអាចដំណើរការឬ ប្រតិបត្តិជាមួយនឹង source file ដោយផ្ទាល់បានឡើយ។ ដើម្បីដំណើរការកម្មវិធីមួយលើប្រព័ន្ធកុំព្យូទ័រណាមួយ នោះ source file ត្រូវតែប្រែជាលេខប្រព័ន្ធគោលពីរ ដែល

កុំព្យូទ័រអាចយល់បាន។ ដំណើរការនេះបង្កើតបាន object file មានលក្ខណៈកម្រិតកណ្តាល ដោយមាន extension .obj ដែលមានន័យថា object។

៦.៣.៣- អ្វីទៅ linking នៃកម្មវិធី C មួយ ?

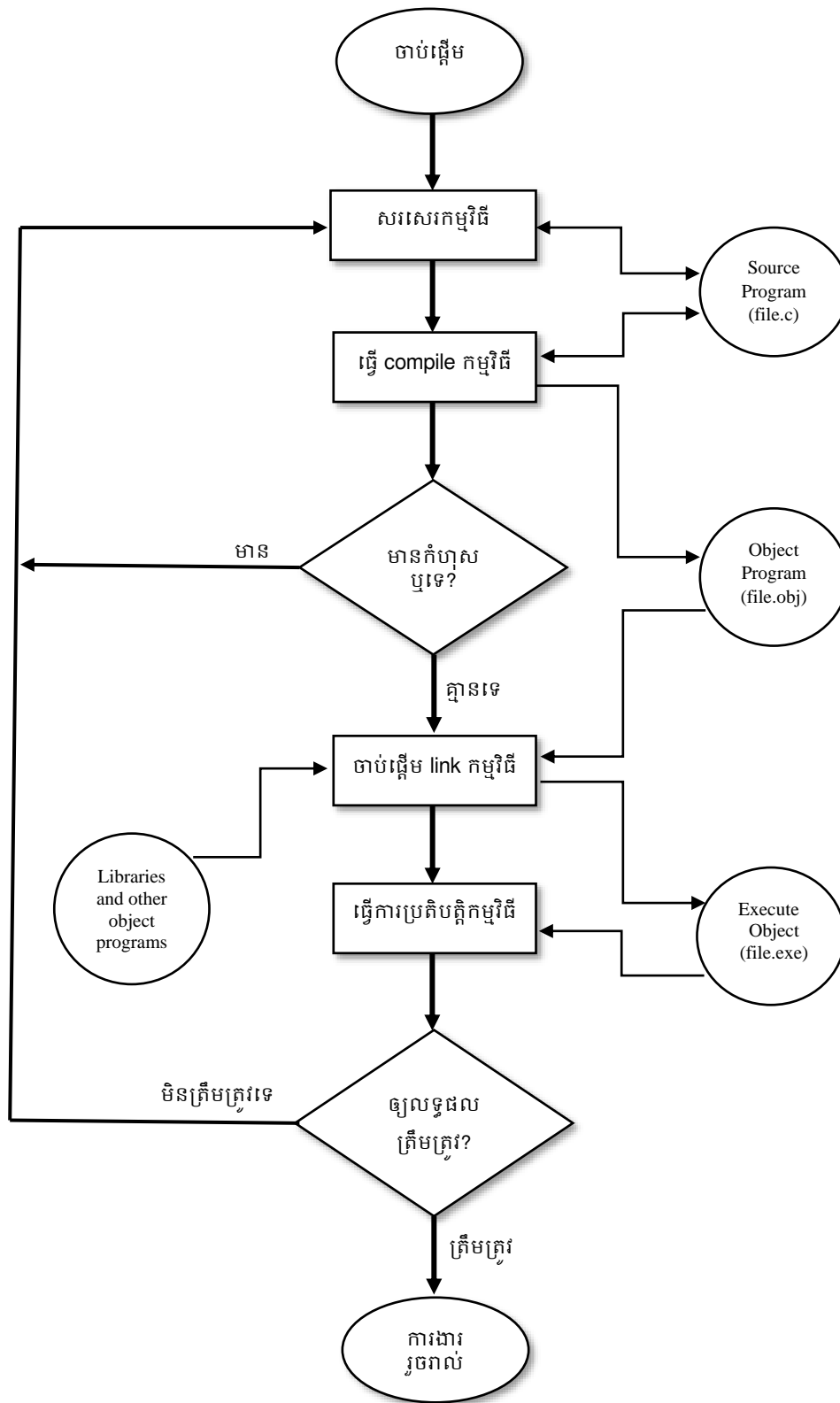
ជាដំណាក់កាលមួយដែលអ្នកបង្កើត compiler ក្នុងគោលបំណងឲ្យអ្នកសរសេរកម្មវិធី C អាចបញ្ចូលការងារផ្សេងៗជាច្រើនបន្ថែម ដូចជា អនុគមន៍ mathematics ដ៏សំបូរ ជាច្រើនពី library។ ក្នុងភាសា C, អនុគមន៍ input និង output ដែលបានផ្ទុកក្នុង library (stdio.h) ហេតុនេះកម្មវិធី C ភាគច្រើនត្រូវការអនុគមន៍របស់ library។ បន្ទាប់ពីធ្វើការ link រួចហើយ វានឹងបង្កើតបាន file មួយដែលមាន extension .exe ដែលអាចដំណើរការបាន។

៦.៣.៤- អ្វីទៅការប្រតិបត្តិ (execution) នៃកម្មវិធី C មួយ ?

Text editor បង្កើតបាន source file ដែលមាន extension .c បញ្ជូនទៅកាន់ compiler បង្កើតបាន object file (.obj) រួចហើយបញ្ជូនទៅកាន់ linker ដែលបង្កើតបាន file .exe។ file ដែលមាន extension .exe នេះ អាចដំណើរការនៅលើកម្មវិធីផ្សេងទៀត ដូចជាលើ DOS ដោយគ្រាន់តែវាយ ឈ្មោះ file នោះនៅលើ DOS prompt។

កំណត់សម្គាល់ ៖

- បើអ្នកចង់បង្កើតកម្មវិធីភាសា C, ធ្វើការ compile និងដំណើរការកម្មវិធីដោយខ្លួនឯង, អ្នកត្រូវតែបញ្ចូល C compiler ទៅក្នុងម៉ាស៊ីនកុំព្យូទ័រ។ ពេលនោះអ្នកអាចប្រតិបត្តិកម្មវិធី C របស់អ្នកបានហើយ។
- ប្រសិនបើអ្នកមិនចង់បញ្ចូល C/C++ compilers នៅក្នុងម៉ាស៊ីនកុំព្យូទ័ររបស់អ្នក អ្នកអាចចូលទៅប្រើ online compilers ដែលនឹងធ្វើការ compile និងប្រតិបត្តិភាសាសរសេរកម្មវិធី C/C++ និងភាសាសរសេរកម្មវិធីដទៃផ្សេងៗទៀតជាច្រើនហើយបង្ហាញលទ្ធផលនៅលើអេក្រង់។



គំនូសតាងឲ្យជំហានសំខាន់ៗសម្រាប់វាយបញ្ចូលកម្មវិធី,
ធ្វើ compile និងប្រតិបត្តិកម្មវិធី C មួយ

៧- ទម្រង់នៃភាសាសរសេរកម្មវិធី C

ទម្រង់នៃកម្មវិធី C មួយត្រូវបានកំណត់ដោយសំណុំគោលការណ៍ច្រើន ហៅថា Protocol ដែលត្រូវអនុលោមតាម អ្នកសរសេរកម្មវិធី នៅពេលសរសេរកម្មវិធី C។ គ្រប់កម្មវិធី C ទាំងអស់ត្រូវមានផ្នែកសំខាន់ៗដូចដែលបានអធិប្បាយខាងក្រោមនេះ៖

- Documentation section (ជាផ្នែកសរសេរបញ្ជាក់ព័ត៌មានរបស់ឯកសារនេះ)
- Link Section (ជាផ្នែកបញ្ជាក់ការតភ្ជាប់ទៅកាន់ទីតាំងនៃឯកសារ)
- Definition Section (ជាផ្នែកកំណត់តម្លៃ)
- Global declaration section (ជាផ្នែកប្រកាសយកទំហំទីតាំង)
- Function prototype declaration section (ជាផ្នែកប្រកាស function)
- Main function (ជាផ្នែកចាប់ផ្តើមដំណើរការកម្មវិធី)
- User defined function definition section (ជាផ្នែកដែលអ្នកសរសេរកម្មវិធីកំណត់ការប្រតិបត្តិរបស់ function)

ឧទាហរណ៍នៃការសរសេរកម្មវិធីកុំព្យូទ័រ ដើម្បីប្រៀបធៀបនូវផ្នែកទាំងអស់ដែលបានអធិប្បាយខាងលើនេះ។ យើងអាចធ្វើការប្រៀបធៀបនូវរាល់ផ្នែកសំខាន់ៗរបស់កម្មវិធី C ខាងក្រោមនេះ៖

```
/*
    Documentation section
    C programming basics & structure of C programs
    Author: Chi Kuong
    Date : 11/11/2016
*/

#include <stdio.h>    /* Link section */

int total = 0;    // Global declaration, definition section

int sum (int, int);    /* Function declaration section */

int main ()        /* Main function */
{
    printf ("This is a C basic program \n");
    total = sum (1, 1);
    printf ("Sum of two numbers : %d \n", total);
}
```



```

    return 0;
}
int sum (int a, int b) /* User defined function */
{
    return a + b;      /* definition section */
}

```

លទ្ធផលទទួលបានពីកម្មវិធីខាងលើ គឺ៖

```

This is a C basic program
Sum of two numbers : 2

```

សេចក្តីអធិប្បាយចំពោះផ្នែកនិមួយៗរបស់កម្មវិធីភាសា C៖

- ចូរយើងសង្កេតមើលលើផ្នែកនិមួយៗរបស់កម្មវិធី C ដោយលម្អិតខាងក្រោមនេះ៖
- សូមកត់សម្គាល់ថា ភាសា C អាចមិនមានផ្នែកដែលបានអធិប្បាយទាំងអស់ខាងក្រោមនេះលើកលែង main function និង Link section។
- ទម្រង់ភាសា C ក៏អាចមិនតាមលំដាប់ដូចបានអធិប្បាយខាងក្រោមនេះដែរ។

Sections	សេចក្តីអធិប្បាយ
Documentation section	គេអាចសរសេរសេចក្តីអធិប្បាយទៅឲ្យកម្មវិធីមួយនូវការបង្កើតឬ ថ្ងៃខែឆ្នាំដែលបានកែប្រែ ឈ្មោះអ្នកនិពន្ធ -ល- នៅក្នុងផ្នែកនេះ។ តួអក្សរ ឬពាក្យឬអ្វីក៏ដោយ ដែលត្រូវដាក់នៅចន្លោះសញ្ញា "/*" និង "*/", ហើយមិនត្រូវបានគិតដោយ C compiler សម្រាប់ដំណើរការ compilation។ សេចក្តីអធិប្បាយទាំងនេះត្រូវរំលងដោយ C compiler ក្នុងពេលធ្វើ compilation។ ឧទាហរណ៍ : /* comment line1 comment line2 comment 3 */
Link Section	Header files ដែលត្រូវការប្រតិបត្តិកម្មវិធី C នោះត្រូវបញ្ចូលក្នុងផ្នែកនេះ
Definition section	នៅក្នុងផ្នែកនេះ អញ្ញាតត្រូវបានកំណត់និងតម្លៃត្រូវដាក់ឲ្យអញ្ញាតទាំងនេះ។

Global Declaration section	អញ្ញាតប្រភេទ global ត្រូវបានកំណត់នៅក្នុងផ្នែកនេះ។ កាលណាអញ្ញាតមួយត្រូវបានប្រើទាំងកម្មវិធី ហើយត្រូវកំណត់នៅក្នុងផ្នែកនេះ។
Function prototype declaration section	Function prototype ផ្តល់នូវព័ត៌មានជាច្រើនអំពី function មួយដូចជា return type, ឈ្មោះប៉ារ៉ាម៉ែត្រដែលបានប្រើនៅខាងក្នុង function។
Main function	រាល់កម្មវិធីភាសា C ត្រូវចាប់ផ្តើមពី function ហើយ function នេះមាន sections សំខាន់ៗពីរ ហៅថា ផ្នែកប្រកាស (declaration section) និងផ្នែកប្រតិបត្តិ ។
User defined function section	អ្នកប្រើប្រាស់អាចកំណត់ functions របស់ខ្លួននៅត្រង់ផ្នែកនេះដែលធ្វើការងារពិសេសដូចជាការងារពិសេសសម្រាប់តម្រូវអ្នកប្រើប្រាស់ម្នាក់ៗ។

៨- ការប្រើអនុគមន៍ printf()

- Functions printf() និង scanf() គឺជា functions ដែលមានស្រាប់របស់ library ក្នុងភាសា C ហើយអាចប្រើបានតាមធម្មតា។ functions ទាំងនេះត្រូវប្រកាសនិងមានទំនាក់ទំនងនឹង macros ដែលបានកំណត់នៅក្នុង "stdio.h" និងជា header file ។
- យើងត្រូវតែបញ្ចូល "stdio.h" file ដូចបានបង្ហាញក្នុងកម្មវិធី C ខាងក្រោម ដើម្បីឲ្យឃើញមានប្រយោជន៍នៃការប្រើ functions printf() និង scanf() ទាំងនេះ។
- printf() function ត្រូវបានប្រើសម្រាប់បង្ហាញ តម្លៃជា ចំនួនគត់ ទសភាគ តួអក្សរ លេខគោល ៨ និងគោល ១៦ នៅលើអេក្រង់។
- គេប្រើ printf() function ជាមួយទម្រង់ %d ដើម្បីបង្ហាញតម្លៃនៃអញ្ញាតចំនួនគត់។
- ដូចគ្នានេះដែរ គេប្រើ printf() function ជាមួយទម្រង់ %c សម្រាប់បង្ហាញតួអក្សរ, %f សម្រាប់អញ្ញាតទសភាគ, %s សម្រាប់អញ្ញាតជា string, %lf សម្រាប់ប្រភេទ double និង %x សម្រាប់អញ្ញាតប្រភេទលេខប្រព័ន្ធគោល ១៦។
- ដើម្បីប្រើ តួអក្សរចុះដើមបន្ទាត់ គេប្រើ "\n" ក្នុង printf() នៃភាសា C ។

កំណត់សម្គាល់ ៖

ភាសា C មានលក្ខណៈប្រកាន់តួអក្សរតូចធំ។ ឧទាហរណ៍ printf() និង scanf() ខុសពី Printf() និង Scanf() ។ គ្រប់តួអក្សរទាំងអស់ក្នុង functions printf() និង scanf() ត្រូវតែមានលក្ខណៈជាអក្សរតូចទាំងអស់។

ឧទាហរណ៍ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការប្រើ function printf() ៖

[illegible]

```

    getch();
    return 0;
}

```

លទ្ធផលទទួលបាន គឺ៖

```

Character is A
String is RoyalUniversityOfPhnomPenh
Float value is 10.234000
Integer value is 150
Double value is 20.123456
Octal value is 226
Hexadecimal value is 96

```

គេអាចឃើញលទ្ធផលជាមួយនឹងទិន្នន័យដូចគ្នាដែលត្រូវដាក់ក្នុងសញ្ញា " " របស់ប្លង់ printf ក្នុងកម្មវិធីខាងលើ

- %d ទទួលបានតម្លៃរបស់អញ្ញាតជាចំនួនគត់ integer និងជំនួសដោយអញ្ញាត (no),
- %c ទទួលបានតម្លៃរបស់អញ្ញាតជាតួអក្សរ character និងជំនួសដោយអញ្ញាត (ch),
- %f ទទួលបានតម្លៃរបស់អញ្ញាតជាចំនួនទសភាគ float និងជំនួសដោយអញ្ញាត (flt),
- %lf ទទួលបានតម្លៃរបស់អញ្ញាតជាចំនួនទសភាគ double និងជំនួសដោយអញ្ញាត (dbl),
- %s ទទួលបានតម្លៃរបស់អញ្ញាតជាតួអក្សរ string និងជំនួសដោយអញ្ញាត (str),
- %o ទទួលបានតម្លៃរបស់អញ្ញាតជាចំនួនគត់ប្រព័ន្ធគោល៨ និងជំនួសដោយអញ្ញាតដែលឆ្លើយតបនឹងតម្លៃចំនួនគត់
- %x ទទួលបានតម្លៃរបស់អញ្ញាតជាចំនួនគត់ប្រព័ន្ធគោល១៦ និងជំនួសដោយអញ្ញាតដែលឆ្លើយតបនឹងតម្លៃចំនួនគត់
- \n ទទួលបានតម្លៃរបស់តួអក្សរ newline។

៩- ការប្រើអនុគមន៍ scanf()

- function scanf() ត្រូវបានប្រើសម្រាប់អានទិន្នន័យជាលេខ ឬជា តួអក្សរ ១តួ(character) និងច្រើនតួ (string) ពី keyboard។

- ចូរពិនិត្យកម្មវិធីខាងក្រោមជាឧទាហរណ៍ដែលអ្នកប្រើបញ្ចូល១តួអក្សរ។ តម្លៃនេះត្រូវបានកំណត់ទៅឲ្យអញ្ញាត "ch" រួចហើយបង្ហាញមកលើអេក្រង់
- បន្ទាប់មក អ្នកប្រើបញ្ចូល string មួយ ដោយតម្លៃនេះត្រូវបានកំណត់ទៅឲ្យអញ្ញាត "str" រួចហើយបង្ហាញមកលើអេក្រង់។

ឧទាហរណ៍ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការប្រើ function printf() និង scanf() ក្នុងភាសា C ៖

```
#include <stdio.h>
#include <conio.h>
void main(){
    char ch;    // ការតាងអញ្ញាត ch ជាប្រភេទ char

    char str[100]; // ការតាងអញ្ញាត array ឈ្មោះ str ប្រភេទ char ប្រវែង 100 ធាតុ

    printf("Enter any character \n"); // បង្ហាញអក្សរឈ្មោះក្នុងចន្លោះសញ្ញា "

    scanf("%c", &ch); // ការអានបញ្ចូលមួយតួអក្សរពី Keyboard

    printf("Entered character is %c \n", ch);
                                // បង្ហាញតម្លៃជាតួអក្សរនៅលើអេក្រង់

    printf("Enter any string ( upto 100 character ) \n");
    scanf("%s", &str); // ការអានបញ្ចូលតួអក្សរមួយឈ្មោះពី Keyboard

    printf("Entered string is %s \n", str);
                                // បង្ហាញតម្លៃជាអក្សរមួយឈ្មោះនៅលើអេក្រង់

    getch();
}
```

លទ្ធផលទទួលបាន គឺ៖

```
Enter any character
a
Entered character is a
Enter any string ( upto 100 character )
```

```
Sokha
Entered string is Sokha
```

- ទម្រង់កំណត់ជា %d ត្រូវបានប្រើក្នុង scanf() ហេតុនេះ តម្លៃដែលបានបញ្ចូលត្រូវតែជាចំនួនគត់ប្រភេទ integer និង %s សម្រាប់តម្លៃជាប្រភេទ string។
- សញ្ញា & ត្រូវបានប្រើពីមុខឈ្មោះអញ្ញាត "ch" នៅក្នុងឃ្លា scanf() ដូចជា &ch ។
- វាដូចទៅនឹង pointer មួយដែលត្រូវបានប្រើសម្រាប់ point ទៅកាន់អញ្ញាត។

90- ការប្រើ Comment

នៅក្នុងភាសា C គេអាចដាក់ comment នៅក្នុង source code របស់គេដែលវាមិនមែនជាផ្នែកមួយនៃកម្មវិធី ហើយត្រូវប្រតិបត្តិឡើយ។ comment ធ្វើឲ្យ source code កាន់តែច្បាស់និងងាយយល់ដល់អ្នកអាន source code មុននឹងយកទៅធ្វើ compile រកកំហុស។ comments មានសារៈសំខាន់ណាស់នៅក្នុងគម្រោងធំដែលមាន source code រាប់រយរាប់ពាន់បន្ទាត់។ កាលណា compiler បានប្រទះ comments នេះ វាមិនធ្វើការចាប់កំហុសក្នុង comments ឡើយ។

Comments មានពីរទម្រង់ គឺ៖

```
/* comment goes here */
```

ឧទាហរណ៍៖

```
#define AGE 6      /* This constant is called AGE */
```

```
/*
 * comment goes here
 */
```

ឧទាហរណ៍៖

```
/*
 * Author: Ly sokun
 * Purpose: To show a comment that spans multiple lines.
 * Language: C
 */
```

និង `// comment goes here`

ឧទាហរណ៍៖

```
#define AGE 6        // This constant is called AGE
```

១១- អំពី escape sequence

Escape sequences ត្រូវបានប្រើនៅក្នុងភាសាសរសេរកម្មវិធី C និង C++ ហើយនិងភាសាផ្សេងទៀតជាច្រើន ដូចជា ភាសា Java និង C#។ Escape sequence មួយ គឺជាលំដាប់អក្សរបន្តបន្ទាប់គ្នាដែលមិនតាងឲ្យខ្លួនវានៅពេលប្រើខាងក្នុងតួអក្សរមួយតួឬ string ប៉ុន្តែវាត្រូវបានប្រែទៅជាតួអក្សរមួយផ្សេងទៀតដែលពិបាកឬមិនអាចតាងឲ្យតួអក្សរដោយផ្ទាល់។

តារាងខាងក្រោមបង្ហាញ escape sequences ដែលបានកំណត់ដោយភាសា C ។

Escape sequence	អត្ថន័យនៃការតាងឲ្យតួអក្សរ
<code>\a</code>	ឲ្យសម្លេង beep
<code>\b</code>	លុបមកខាងឆ្វេងមួយតួអក្សរ
<code>\f</code>	កំណត់ទំហំ formfeed
<code>\n</code>	ចុះដើមបន្ទាត់
<code>\r</code>	ត្រឡប់ទៅដើមបន្ទាត់
<code>\t</code>	ដកឃ្លាមួយ tab តាមជួរដេក
<code>\v</code>	ដកឃ្លាមួយ tab តាមជួរឈរ
<code>\\</code>	បង្ហាញតួអក្សរ \
<code>\'</code>	បង្ហាញតួអក្សរ '
<code>\"</code>	បង្ហាញតួអក្សរ "
<code>\0</code>	តាងឲ្យតួអក្សរ null

សំណួរនិងលំហាត់

១- ចូរនិយាយពីប្រវត្តិភាសា C។

២- ចូរនិយាយពីលក្ខណៈពិសេសរបស់ភាសាសរសេរកម្មវិធី C។

៣- ចូរនិយាយពីអត្ថប្រយោជន៍នៃភាសាសរសេរកម្មវិធី C។

៤- តើភាសាសរសេរកម្មវិធី C ត្រូវបានចាត់ទុកជាភាសាក្នុងកម្រិតណា?

៥- តើគេចែកភាសាសរសេរកម្មវិធីជាប៉ុន្មានលំដាប់? ចូរអធិប្បាយ។

៦- ចូរនិយាយពីជំហានសំខាន់ៗនៃការសរសេរកម្មវិធី C រហូតដល់បង្ហាញលទ្ធផលនៅលើអេក្រង់។

៧- ចូរនិយាយពីមុខងាររបស់អនុគមន៍ scanf() និង printf()។

៨- ចូរសរសេរកម្មវិធីមួយដែលអាចឲ្យគេគណនាក្រឡាផ្ទៃនិងបរិមាត្រនៃចតុកោណកែងមួយ កាលណាគេបញ្ចូលតម្លៃនៃប្រវែងទទឹង និង បណ្តោយពី keyboard។

៩- ចូរសរសេរកម្មវិធីមួយដែលអាចឲ្យគេបញ្ចូលតម្លៃពី keyboard និងបង្ហាញលទ្ធផលនៅលើអេក្រង់ មានទម្រង់ដូចខាងក្រោមនេះ៖

Name : Sopheap

Sex : F

Age : 23

Total score : 9.75

១០- ចូរសរសេរកម្មវិធីមួយដែលអាចឲ្យគេបញ្ចូលតម្លៃពី keyboard ដើម្បីគណនាផលបូកចំនួនស្វ័តពី 1 ដល់ n ដែលមានរូបមន្តដូចខាងក្រោម៖

ក. $S_n = n(n+1)/2$

ខ. $S_n = n(n+1)(2n+1)/6$

គ. $S_n = n^2(n+1)^2/4$

