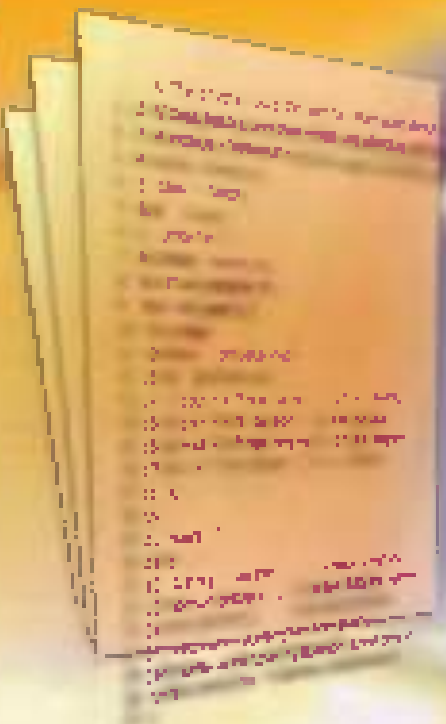


အိမ်ထောင်
ကျမ်းကျမ်း



C + +

Complete

AUNG MYINT (M.E., AUSTRALIA)

CHAPTER 1

C++ BASICS

1.1.	A Brief History of C and C++	10
1.2.	My First C++ Program	12
1.3.	Identifiers	15
1.4.	Keywords	16
1.5.	Variables	17
1.6.	Constants	24
1.7.	Expressions	30
1.8.	Assignments	31



Complete

CHAPTER 2

FUNCTIONS

2.1.	Create a Simple Function	46
2.2.	Pass the Variables by Value	50
2.3.	Pass Variables by Reference	59
2.4.	Default Function Arguments	63
2.5.	Inline Functions	65
2.6.	Local and Global Variables	65
2.7.	Overloaded Functions	69
2.8.	Recursion	73
2.9.	Passing Structure Variables	77



Complete

CHAPTER 3

PROGRAM FLOW CONTROL

3.1.	The if Statement	82
3.2.	The goto Statement	86
3.3.	The if-else Statement	92
3.4.	Nested if Statement	96
3.5.	The while Statement	101
3.6.	The do-while Statement	108
3.7.	The for Statement	114
3.8.	Nested Loops	120
3.9.	The switch Statement	123
3.10.	The break Statement	127
3.11.	The continue Statement	129



Complete

CHAPTER 4

POINTERS AND ADDRESSES

4.1.	Referencing Variables with Pointers	135
4.2.	Pointers and Arrays	136
4.3.	Pointers to Structures	140
4.4.	Pointers as Function Arguments	142
4.5.	Passing Arrays by Pointers	145
4.6.	Pointers and Strings	151
4.7.	Arrays of Pointers to Strings	156
4.8.	Pointers to Pointers	158
4.9.	Memory Allocation	164
4.10.	Using Reference Variables	158



Complete



1979 ခုနှစ် နိုဘဲလ်ဆုရှင်ဟောင်း စတင်ဖွဲ့စည်းပြင်ဆင်သူ New Jersey (ပြည်ထောင်စု) AT & T Bell Laboratories မှာလုပ်ကိုင်နေတဲ့ Bjarne Stroustrup ဆိုသူကွန်ပျူတာပရိုဂရမ်မာတင်ဖို့က C language ဝါဒီဟာဖြစ်ပြီး C++ ဆိုတဲ့ object oriented programming language တစ်ခုကိုရေးသားခဲ့ပါသည်။ ဒါဟာ C++ မဟုတ်ပါဘူး။ C with Classes ဝါဒီဟာလည်းနောင်ကပါ။ 1983 ခုနှစ်ရောက်မှ C++ သို့မဟုတ်ပြောင်းလဲလိုက်သည်။ C++ က C ကိုကဆင့်မြှင့်တင်တဲ့ version ပါ။ C++ source code ဆွဲကို C compiler မယူပေါ်မှာ run ခဲ့ရမယ်။ C front ဆိုတဲ့ interpreter ကိုမယူမဖြစ်ပေမယ့်။ တစ်ချို့က C++ ကို C ရဲ့ superset လို့ မြင်ကြသည်။ C language မှာ လုပ်လို့မရတာတွေနဲ့လက်တွေ့ကို C++ မှာကလည်း လုပ်လို့ရတာကို။ ယင်းပေးခဲ့ရသည်။ C++ language မှာကဆင့်သင်မြှင့်တင်တဲ့ keyword ဆွဲက C language မှာ မပါဘူးဆိုတာကို ပါပဲ keyword မွေကို C program မှာ identifier မွေပေးမနေ ဟုဆိုပြောနိုင်တာပါ။ ဒါ့ကို C program မှာပဲ C++ က compile မှုပေးလို့ရတာ။ C compiler ဆိုတဲ့ compile လုပ်ပြီး run ပေးနိုင်မှာပါ။ တည်ငြိမ်ခြင်း တစ်ခုကလေးနဲ့ C language လိုပဲဆိုပြောတာဆိုရင် C++ ကိုဆက်လက်ရေးသားတဲ့အခါ လွယ်လွယ်ရေးသားမှာပါ။

A Chronological C and C++

1986 Cfront ကို MS-DOS မှာတင်ဆက်တယ်။ MS-DOS C compiler ကိုရဖို့ Microsoft ကနေထွက်ခွင့်ရခဲ့ပါတယ်။

1987 Borland ကနေ ထုတ်ဝေတဲ့ Turbo C (for MS-DOS) မှာ ကမ်းခြား version မထွက်ခင်ပိုမိုကောင်းမွန်ခဲ့ပါတယ်။

1989 C++ standardization မကျဘဲ ANSI က တောင်ဆက်ပေးဖို့ဖြုန်းပါတယ်။

1990 C standard document ကို ANSI ကနေ ပုံနှိပ်ထုတ်ဝေခဲ့ပါတယ်။ Borland ကနေ C++ compiler (for MS-DOS) ကိုဖြန့်ချိပါတယ်။

1991 Microsoft ကနေ C compiler (for MS-DOS) ကို ဖြန့်ချိပါတယ်။

1998 ANSI နဲ့ ISO တို့ကနေ C++ standard definition document ကို ကန့်သတ် ဝန်ဆောင်မှုထုတ်ဝေခဲ့ပါတယ်။

၁.၂ My First C++ Program

၁။ (၁-၁) ကုဒ်အဖြစ်ကို `Ex101.cpp` program မှာဆိုရင် လွှန်ပတ်တဲ့ဖောင်မထုတ်ဖို့ကို C++ program `g++Ex101.cpp` program က `g++` ခေတ် `run` လို့လုပ်တယ်။

```

Ex101.cpp
// Listing 1.1 My First C++ program

#include <string>

int main()
{
    // Display a message on the screen
    cout << "My first C++ program" << endl;

    return 0;
}

```

Figure 1.1

3. program ကို Compile ပြုပြီး Run ပြုလုပ်ရန် - ရှေးရိုးတော်တော်များ My First C++ program ခေါ်တွင်ပြီး ငါ့ နယ်လောက်ကို အညွှန်ထည့်ပြီး Figure 1.1 ကို ကြည့်ပါ။

```

Quincy 99
My First C++ program
Press any key to return to Quincy...

```

Figure 1.2

4. program ကို လေ့လာတဲ့ ဖြစ်ရပ်အသစ်များ

- ကြေးမုံကုမ္ပဏီ (Listing 1.1) My First C++ program ခေါ်တဲ့ statement များမှာ double-slash characters // ကို single-line comment ကို ခေါ်ပုံတည်း။ မိမိလေ့လာတာတွေမှာ program ကလေးတိုတိုတိုတိုတို့ ထည့်ရတာတွေအတွက်ပါ။ PROGRAM ကို

comment သည် ပုံသဏ္ဍာန်မှ သွေးမှ သွေးမှ ပုံသဏ္ဍာန်အတိုင်း ဖော်ပြ၍ program ၏ တွင်ပေး
 သည့်အပြင် ရှိသည့်အချက်အလက်များကို program ၏ run ဝင်ပေးရာတွင် white space သာမရှိ
 effect ရှိပုံနှင့် အင်္ဂလိပ် comment မှ စတင်ပေးခြင်းအတိုင်းမူ comment line
 သွေးကို သွေးကို " /* */ format မှိုတိုင်းပေးသည့် ဝါးတိုင်းမှိုတိုင်းအတိုင်း compiler
 တွင်ပေးသည်။

- C++ သည် free-form language အဖြစ်အရည်ပြင်းပေးပြီး program ၏ compile နှင့်
 တွင်ပေးရာ new lines spaces tabs blank lines အစရှိသည့် white-space character
 သွေးကို သွေးကိုသွေးကို သွေးကိုခွဲခြားခြင်း compilation မှ သွေးကို သွေးကိုခွဲခြားခြင်း
 တွင်ပေးသည်။ စတင်ပေးသည့် program listing မှ white space သွေးကို သွေးကိုခွဲခြားခြင်း
 source code သွေးကို သွေးကိုခွဲခြားခြင်းအတိုင်း ပေး၍ Listing 1.3 ၏ သွေးကိုခွဲခြားခြင်းအတိုင်း
 program ၏ run ဝင်ပေးသည့် program code သွေးကို သွေးကိုခွဲခြားခြင်းအတိုင်း ပေးသည်။

```
#include <iostream>
int main() { // Display a message on the screen
cout << "My First C++ program"; return 0; }
```

- #include <iostream> statement ၏ preprocessing directive ရှိသည့် ဝါးတိုင်း
 current program မှ <iostream> header ရှိသည့်အတိုင်း #include directive သွေးကို
 compiler ၏တိုင်းခြားခြင်းအတိုင်း console output သွေးကို သွေးကိုခွဲခြားခြင်း
 functions နှင့် global value သွေးကို <iostream> standard library header မှိုတိုင်း
 သွေးကိုခွဲခြားခြင်းအတိုင်း ဝါးတိုင်းခွဲခြားခြင်း cout ၏ program မှိုတိုင်းခြားခြင်းအတိုင်း
 <iostream> မှိုတိုင်း angle brackets မှိုတိုင်း သွေးကိုခွဲခြားခြင်းအတိုင်း header မှိုတိုင်း
 compiler system တွင် supply ရှိသည့်အတိုင်း ဝါးတိုင်းခွဲခြားခြင်းအတိုင်း header ရှိသည့်
 include သွေးကို သွေးကိုခွဲခြားခြင်းအတိုင်း bracket separate quote (" ") သွေးကို
 သွေးကိုခွဲခြားခြင်းအတိုင်း #include "myheader.h" အတိုင်းခွဲခြားခြင်း။

- int main() statement အတိုင်း main() function သွေးကို integer(int) value သွေးကို
 ၏ return ပေးသည့်အတိုင်း ဝါးတိုင်းခွဲခြားခြင်း main() function မှိုတိုင်း
 သွေးကို သွေးကိုခွဲခြားခြင်းအတိုင်း argument မှိုတိုင်း ဝါးတိုင်းခွဲခြားခြင်း။

Non-Identifiers

asm	else	operator	throw
auto	enum	private	true
bool	explicit	protected	try
break	extern	public	typedef
case	false	register	typeid
catch	float	reinterpret_cast	typename
char	for	return	union
class	friend	short	unsigned
const	goto	signed	using
const_cast	if	sizeof	virtual
continue	inline	static	void
default	int	static_cast	volatile
delete	long	struct	wchar_t
do	mutable	switch	while
double	namespace	template	
dynamic_cast	new	this	

2.9 Variables

variable \Rightarrow program can store data value \Rightarrow memory location \Rightarrow storage class \Rightarrow scope

variable ဆိုရာ type name ပတ်နဲ့ ခြိမ်ပါသည်။ type ဆိုတာ variable ရဲ့ format နဲ့ data ပေးပေးပုံကို ဆက်ဆံပေးတာပါ။ variable ကို ချဲ့ကန့်ဖို့တွေကို program ထဲမှာ ခြိမ်ပေးပေးခြင်းပင်ဖြစ်ပေမည်။ ချဲ့ကန့်ပေးပုံကို data type ကောက်ချဲ့ခြင်းလို့လည်း type declaration မှာ ခြေငြိတ်ထားပေးတိုင်း ခြေငြိမ်းပေးပါ။ C++ မှာကလေး data type တွေက bool, char, wchar_t, int, float နဲ့ double ခြေငြိမ်းပါတယ်။ variable type တွေကို အပို ခြိမ်ပေးနိုင်ရန် unsigned, long, short, long double ခြေငြိမ်းပေးနိုင်ပါသည်။

The bool Type

bool variable သည် true ဆိုသည့်ကိစ္စနှင့် false ဆိုသည့်ကိစ္စကို value (1) ချဲ့ကန့်ပေးခြင်းပင်။ boolean (logical) type ကို အသုံးပြုပါ။ arithmetic expression တစ်ခုခုကို bool variable ကို အသုံးပြုပေးမည့်အခါ expression ကို true ခြေငြိမ်းပေးပါ။ integer value 1 ကို return ပေးပေးနိုင်ခြင်းဖြစ်ပြီး false ဆိုသည့်ကိစ္စကို return ပေးပေးပါ။ တစ်ကောင်လုံး integer ကိုလည်း bool ခြေငြိမ်းပေးပေးခြင်းပင်ဖြစ်ပေမည်။ integer 0 ကို false ခြေငြိမ်းပြီး integer nonzero ကို true ခြေငြိမ်းပါ။ (3, ၄) ကိုလည်း true ခြေငြိမ်းပါ။ C/C++ program တစ်ခုကို bool variable အသုံးပြုပေးခြင်းကို အောက်ပါအတိုင်း ဆောင်ရွက်ကြည့်ပါ။

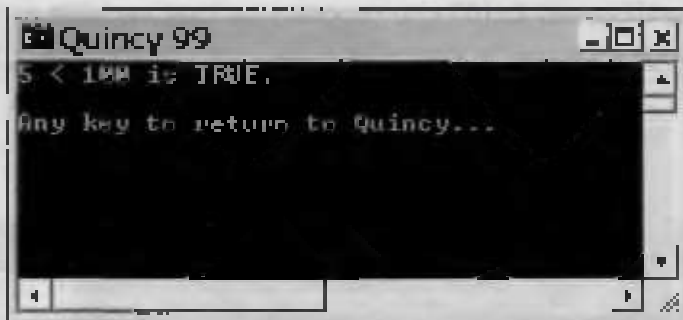
```

1 # Using I/O Using a bool variable
2
3 #include <iostream>
4
5 int main()
6 {
7     bool flag; // bool variable
8
9     flag = 3 < 100; // Set logical value to flag
10    if (flag)
11        cout << "The flag is TRUE!" << endl;
12
13    return 0;
14 }

```

၅ (၁၂၅)

၂ program ကို run လိုက်တာတွေနဲ့ပစ္စည်းကောင်းမှာ $5 < 100$ is TRUE. ဆိုတဲ့စာကြောင်း ပေါ်လာတာကိုတွေ့ရပါမိမ့်မယ်။ ၆ (၁-၄) ကိုကြည့်ပါ။



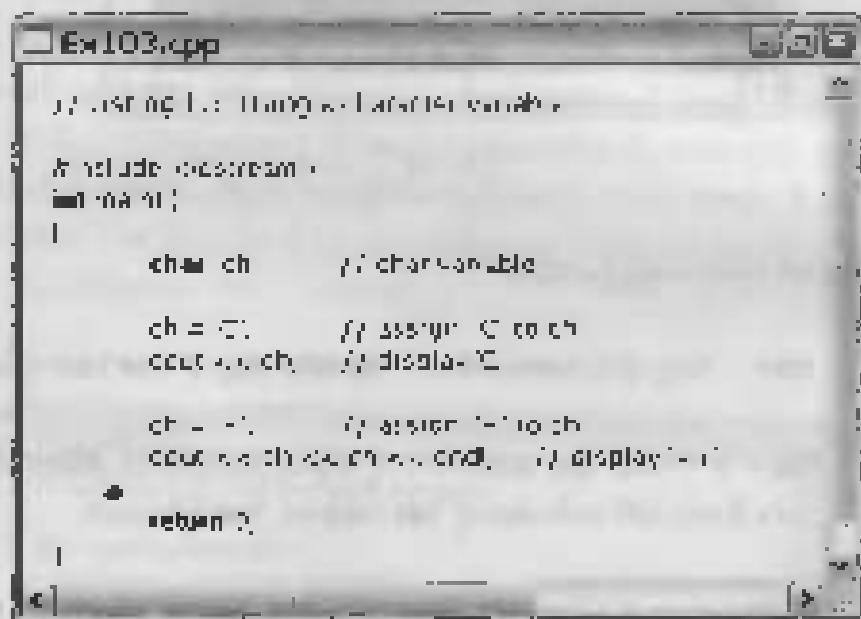
၆ (၁-၄)

၃ program ကိုလေ့လာကြည့်မယ်ဆိုရင်

- `bool flag;` ဆိုတဲ့ statement ဟာ identifier `flag` ကို type `bool` လို့ ခြေကြားပေးပါ။
- `flag = 5 < 100;` ဆိုတဲ့ statement ကို logical assignment statement ယူခေါ်ပါလိမ့်။ $5 < 100$ ကိုကယ်တာမို့ `flag` ကိုပေးတာ True ပြန်ပါတယ်။
- `if (flag)` ဆိုတာ `if (True)` ဟာလို့ခေါ်တာပေါ့။ True ပြန်တဲ့အတွက် `cout << "5 < 100 is TRUE."` ဆိုတဲ့ statement ကို execute လုပ်ပြီး console (screen) မှာ `5 < 100 is TRUE.` ဆိုတဲ့ string constant ကို display လုပ်ပျက်တယ်။ တစ်ကယ်တော့ `flag = 5 > 100` ဆိုရင် False ပြန်တဲ့အတွက် `if (flag)` နောက်က statement ကိုမပြုလုပ်ဘဲ တော့မသွားတော့ပါ။
- နောက်ဆုံးမှာ `return 0;` statement ကိုတွေ့ရင် `main()` function မှာအလုပ်ပြီးသွားပြီမို့ constant integer (0) value ကို operating system ဆီ return လုပ်ပေးပါလိမ့်မယ်။ ပြီးတော့ရင် right brace `{ }` ; ဝေလင်တကနေ program ကိုဆုံးပြန်တင်တာကို ခြေကြားပေးပါလိမ့်။

The char Type

■ char variable သို့မဟုတ် 8-bit bytes ASCII character set ထဲက character value ရေးတဲ့ သို့မဟုတ် ပုံမှန် char identifier တစ်ခုကို type char နဲ့ declare ပြုလုပ်ထားတဲ့ char ch, သို့မဟုတ် မူလကဲ့သို့ (၁.၅) နဲ့ (၁.၆) ကဲ့သို့ (၅.၁) Ex104.cpp program မှာ char variable တွေကို ပြုလုပ်ထားတာတွေကို ကြည့်ပါ။ Ex103.cpp program ကို run လုပ်ထားတဲ့ ရုပ်ပုံများကို (၁.၆) နဲ့ (၁.၇) ကဲ့သို့ ပြောနိုင်ပါသည်။



```
1 // using the third character variable
2
3 #include <iostream>
4 int main()
5 {
6     char ch; // char variable
7
8     ch = 'C'; // assign 'C' to ch
9     cout << ch; // display 'C'
10
11    ch = 'H'; // assign 'H' to ch
12    cout << ch << " and " << ch << endl; // display "H and C"
13
14    return 0;
15 }
```

(၁.၆)

■ (၁.၆) မှာ Ex103.cpp program ကို run ပြုလုပ်ထားတဲ့ ရုပ်ပုံများကို (၁.၇) ကဲ့သို့ ပြောနိုင်ပါသည်။ program ကို ထပ်မံထပ်မံ run လုပ်ထားတာတွေကို ကြည့်ပါ။

- နေရာမှာ identifier ch ကို type char သို့မဟုတ် char ch ကို character 'C' နဲ့ assign ပြုလုပ်ထားတာတွေကို (၁.၆) မှာ ch ရဲ့ value ဖြစ်တဲ့ 'C' တွေကို screen မှာ display လုပ်ပေးတာတွေကို ကြည့်ပါ။


```

Ex104.cpp
// Listing 1.4: Using an int variable

#include <iostream>
int main()
{
    int num; // int variable

    num = 12345; // assign 12345 to num

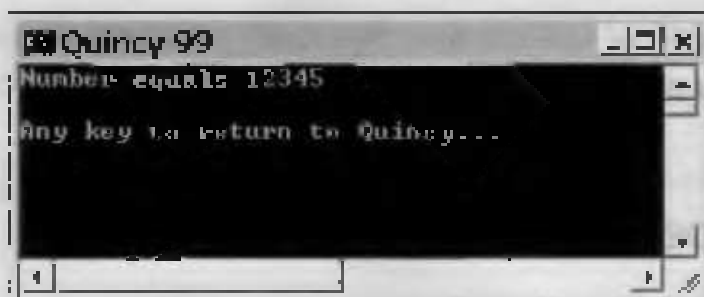
    // display the number
    cout << "Number equals " << num << endl;

    return 0;
}

```

၁.၁၅

Ex104.cpp program ကို num လိုက်နာသည့်အခါတွင် ထွက်ပေးသောရလဒ်မှာ Number equals 12345 ခို့
 ပေါ်လာသည်ကိုး ၁.၁၅ ကိုကြည့်ပါ။



၁.၁၆

ဒီ program ကိုအလုပ်ပြုနိုင်ရန်

- ဆရာမက identifier num ကို type int လိုက်နာပြုလုပ်ပေးပြီးနောက် num ကို 12345
 သတ်မှတ်ပေးပြီး assign လုပ်ပေးပြီး num မှ value ကို string constant ခေါ်ဆိုထွက်ပေး
 display လုပ်ပြုနိုင်ပေးပါ။

\Gn n	Octal number (nnn)
\a	Audible bell character
\b	Backspace
\f	Formfeed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\x	Hexadecimal number (hhh)
\?	question mark

Integer Constants

Integer constants ဆိုတာ type name မပါဘဲ long ဆိုလျှင် short, signed ဆိုလျှင် unsigned fixed value ကျင့် ဆိုလိုသည်။ integer constant မှတ် (3) မှ: (၁) ထပ်ခွဲကားပါးကောင်း၊ (၂) Decimal constants (base 10) ()၊ Octal constants (base 8) & (၃) Hexadecimal constants (base 16) ဖြစ်ပါ။ decimal constant ကိုသာအချို့က ယူကော (9) ဝေဝေ ကောင်းကျင့်စုဝါးပြင်ကောင်း၊ constant မှတ် (7) ကို ဆိုလျှင် (2) ကိုသာက ဝါးဝေကောင်းကောင်း ကျင့်စုဝါးပြင်ကောင်း၊ ကျင့်စု: constant value ကာ သတ်မှတ်ထားတဲ့ lower bound & upper bound ကြားက ကျင့်စုကျင့်စုပါး။ C++ program က ကျင့်စုကျင့်စု decimal constant ကျင့်စု ကာ 0, 1234, 32768 ကို (၃) ကျင့်စု constant ကျင့်စု decimal constant ကျင့်စု ကျင့်စုပါး။

- 23,456** constant မှတ် comma (,) ပါဝင်ပါးပါး
- 54.68** ကောင်းကျင့်စု ကျင့်စုပါးပါး
- 32 591** space ကျင့်စုကျင့်စုကျင့်စုပါးပါး
- 0987** ကျင့်စုကျင့်စုကျင့်စု ကျင့်စုကျင့်စုကျင့်စုပါးပါး
- 956705** constant value ကာ upper bound | ကျင့်စုကျင့်စုကျင့်စုပါးပါး ကျင့်စု decimal constant ကျင့်စုပါးပါး


```

t×106.cpp
// Listing 1-6. Concatenated string constants
#include <iostream>

int main()
{
    cout << "This is the beginning of a very long message,\n"
          "that spans several lines of C++ programming code.\n"
          "This format allows a programmer to build long\n"
          "string constants without going past the\n"
          "program editor's right margin.\n";

    return 0;
}

```

Figure 1-10

Figure 1-10 shows the output of the program in Listing 1-6. The program concatenates several lines of string constants to produce a single long string constant. The output of the program is shown in Figure 1-11.

```

Quincy 94
This is the beginning of a very long message,
that spans several lines of C++ programming code.
This format allows a programmer to build long
string constants without going past the
program editor's right margin.

Any key to return to Quincy...

```

Figure 1-11

The program in Listing 1-6 demonstrates the use of concatenated string constants.

- string constant `\n` is a newline escape sequence. It is used to create a new line in the output of the program. The output of the program in Listing 1-6 is shown in Figure 1-11.

Enumerated Constants

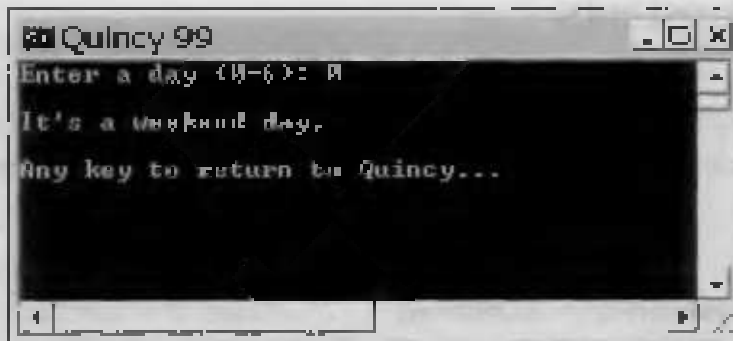
၁။ program တစ်ခုခု ဆက်တိုက်ဆက်တိုက် data set တစ်ခုကို type ကိစ္စရပ်အရ ဆက်တိုက်ဆက်တိုက် enumerated constant ကိုအသုံးပြုကြပါက သူက COLOR ကို enumeration ကိစ္စရပ်အဖြစ်အဖြစ် (RED, BLUE, GREEN, WHITE, BLACK) ဆိုတဲ့အတွက် (5) ဆိုတဲ့ကိစ္စရပ်အဖြစ် enum COLOR {RED, BLUE, GREEN, WHITE, BLACK}; ဆိုတဲ့ statement ကိုရေးရပါက COLOR ဆိုတာက enumeration type name ဝါ RED ဆိုတာက symbolic constant ကို 0 ကိုအသုံးပြုပါက ကိုတာက ၁, 2, 3, 4 ကို အသုံးပြုနိုင်ဖို့ ဆက်တိုက်ဆက်တိုက် ပေးရတဲ့ enum COLOR {RED=0, BLUE=1, GREEN=500, WHITE, BLACK}; ကိုအသုံးပြုနိုင်ဖို့ BLUE = 51, WHITE = 501 & BLACK = 502 ကို enumeration type ကို အသုံးပြုဆက်တိုက်ဆက်တိုက် ပုံ (၁၀.၁၄) မှာပါ ပြဿနာ Ex107.cpp program ကိုရေးပေးကြည့်ပါ။

```
// Listing 1.7 Using an enumerated constant
#include <iostream>

int main ()
{
    enum days {Sun, Mon, Tue, Wed, Thu, Fri, Sat}
    int choice;

    cout << "Enter a day (0-6) : ";
    cin >> choice;
    if (choice == Sun || choice == Sat)
        cout << "You're a weekend day.\n";
    else
        cout << "You're a week day.\n";
    return 0;
}
```

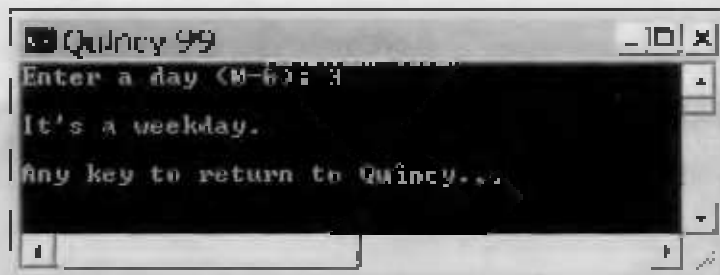
၂။ အခုပဲ program ကို run လိုက်ကြို Enter a day (0-6) : ကို prompt တစ်ခုပေါ်လာတဲ့အခါမှာ ၆ ကိုရိုက်ထည့်ပြီး ENTER နှိပ်လိုက်တာနဲ့ It's a weekend day. လိုပေါ်လာပါလိမ့်မယ် ဝါ (၁၀.၁၄) ကိုကြည့်ပါ။



ပုံ (၁.၈၄)

ဒီ program ကိုရေးတဲ့အကြောင်းအရာ

- အခန်း: identifier days ကို type enum ကိုသုံးပြီး (Sun, Mon, Tue, Wed, Thu, Fri, Sat) ကို assign လုပ်ပေးရမယ်။ identifier choice ကို type int သုံးပြီးရည်ညွှန်းပေးမယ်။
- cout << "Enter a day (0-6) : ", ဆိုတဲ့ statement ကနေ choice ကတော့ data တည်ဆောက်မှု prompt လုပ်ပေးပါကထဲမှာ cin >> choice; ကနေ choice ကတော့ data ကို 0 ကိုရိုက်ထည့်ပေးမယ်ဆိုပါနဲ့။
- ဒါ့နောက် if (choice == Sun || choice == Sat) statement ကနေ choice က Sun (0) ဆိုရင် Saturday (=6) ကနေပေးမယ်ဆိုပါကထဲမှာ 0 ပြန်ထုတ်ရမယ်။ if () မှာပေးမယ်ဆိုရင်မှာ cout << "It's a weekend day.\n"; ဆိုတဲ့ statement ကို execute လုပ်ပေးမယ်ဆိုပါကထဲမှာ ဒီနေရာ ထဲမှာပဲ It's a weekend day. ဆိုပေါ်တင်ပြီးပဲ တစ်တည်း choice ကတော့ data ကို 3 ယူရင်မှာ It's a weekday. ဆို display လုပ်ရမယ်။ ပုံ (၁.၈၅) ကိုကြည့်ရင်။



ပုံ (၁.၈၅)

Address Constants

C++ program များမှာ pointer တွေကိုပေးသုံးကြိုက်သလိုရင် Address constant များကို ကလေးပြု ရတော့မယ်။ variables function တွေမှာ memory address တွေ ခြိမ်ယာသ်။ ဒီ address တွေကို C++ program မှာ ဝေဖတ်ပါးစာကိုး reference လုပ်လို့ရပါတယ်။

```
CounterPtr = &Counter; // address of a variable
FuncPtr = &DoFunction; // address of a function
```

၁.၇ Expressions

expression ဆိုတာ ကလေးထဲထဲပြုမိနေမိတဲ့ constant တွေ variable များ array element တွေ function call တွေကို operator များနဲ့ဆက်စပ်ထားတာပါ။ expression ကလေးကိုပြုရတဲ့အခါမှာပဲ ထပ်ပြီးလုပ်ရတဲ့ return လုပ်ပေးမယ်။ C++ expression တွေကို နမူနာပေးပြီးပြရအောင်ဆိုရင် `count * 10 & 1.8*(degC+32)` ဆိုတာမျိုးပေါ့။

Arithmetic Operators

အကလေးပြု ပါးဆိုရင် operator တွေ ဘယ်နှစ်မျိုးရှိလဲ။ arithmetic operators unary operators relational and conditional operators စတော့ပေသလိုဖြင့် ပေးဆိုထားတော့ ကလေးကိုကြိုတင်ပေးပြီး ထပ်ပါးလုပ်သုံးရင် expression ဆိုသလိုဖြစ်လာတာပါ။ C++ မှာပါးစာ) arithmetic operator တွေကို `math (၁-၄)` မှာပေးပြီး ဝေဖတ်ပါးစာ) ဒီ operator များကိုအသုံးပြုပြီး တွက်လဲပေးမှာပေးတော့မို့လို့လုပ်ရမှာပေါ့။

ဇယား (၁.၄) Arithmetic Operators

Symbol	Description
+	Unary plus

-	Unary minus
*	Multiplication
/	Division
%	Modulus
+	Addition
-	Subtraction

၁.၆ Assignments

၁။ Assignment statement တစ်ခုတည်းခုတ် expression တစ်ခု၏(ပြောင်းလိုက်တဲ့) ရလဒ်ကို return value ကို variable တစ်ခု assign လုပ်ပေးပုံကို ပုံ (၁.၁၆) မှာပင် ပြထားတဲ့ Ex108.cpp program ကို Celsius degree တစ်ခုကို $1.8^{\circ}\text{C} + 32$ ဆိုတဲ့ expression ကို အသုံးပြုပြီး degree Fahrenheit ကိုပြောင်းပေးတဲ့ program ဖြစ်ပါမယ်။

```

Ex108.cpp
// Listing 1.8 Assigning an expression

#include <iostream>
int main()
{
    float cel, fah;

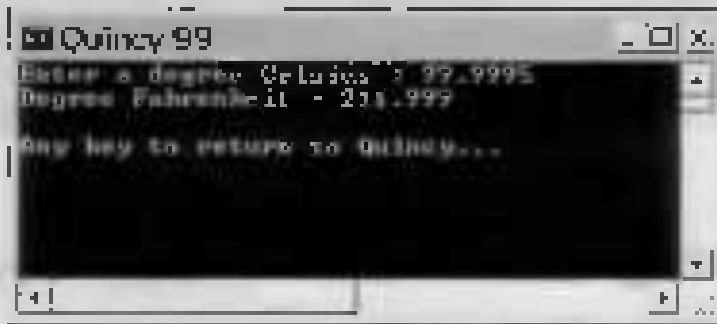
    cout << "Enter a degree Celsius : ";
    cin >> cel;
    fah = 1.8*cel + 32;
    cout << "Degree Fahrenheit = " << fah << endl;

    return 0;
}

```

ပုံ (၁.၁၆)

• `EX03B.cpp` program ကို run လိုက်မည်ဆိုရင် ဤအတိုင်းပုံမှာ Enter a degree Celsius :
 ကို prompt ကနေ ဖတ်ပြီးနောက် 99.9995 ကိုရိုက်ထည့်ပြီး ENTER ခုံထိတ်ရင် Degree Fahrenheit = 211.999 (ပုံနှိပ်ပါသည်) ဖြစ်သည်။



ပုံ (၁-၇)

၁။ ဤ program ကိုလေ့လာကြည့်မည်ဆိုရင်

- မတူရင်း identifier `cel` ကို `cel` ဆိုပြီး `type float` ခြုံငုံပြီး `Enter a degree Celsius` ဆိုတဲ့စာကြောင်း screen ပေါ်လာအောင် `cout << "Enter a degree Celsius" ;` ကိုရေးပါသည်။
- `cin >> cel;` နှင့် `statement` ဆိုတဲ့အတိုင်းကြောင့် `cel` ကို `cel` data ကို user က keyboard ကနေရိုက်ထည့်ပေးရမည်။ `cel` ကို `cel` data ကို 99.9995 ကိုရိုက်ထည့်ပေးပါသည်။
- `cel = 99.9995` ကို program က သိသွားပြီဆိုတာ `fah = 1.8*cel + 32 = 1.8*99.9995 + 32 = 211.999` ဆိုတဲ့စာကြောင်း ပုံနှိပ်ပေး ခဲ့ကြောင်းသိရပါသည်။
- `cout << "Degree Fahrenheit = " >> fah >> endl;` ဆိုတဲ့ `statement` ကနေ နှစ်ခုအား Degree Fahrenheit = 211.999 ဆိုတာကို ပုံနှိပ်ပေးအောင်ရေးပါသည်။ `<< endl; expression` သို့ `text; statement line` ကိုရေးလိုက်ရင် နှစ်ခုပေးပါသည်။


```

Ex109.cpp
// Listing 1.9 Using the relational and logical operators
#include <iostream>
int main()
{
    double a,b,c,area;

    cout << "Enter three sides of triangle\n";
    cin >> a >> b >> c;
    if(a+b > c && a+c > b && b+c > a)
    {
        s = (a+b+c)/2;
        area = sqrt(s*(s-a)*(s-b)*(s-c));
        cout << "Area of triangle = " << area << endl;
    }
    else
        cout << "Triangle can't be formed";

    return 0;
}

```

Figure 1.10

When we run the program, it prompts us to enter three sides of triangle. The prompt asks for three sides: 3 space 4 space 5. After pressing the ENTER key, the output is Area of triangle = 6. If we press any other key, it will display Triangle can't be formed.

```

C:\Quincy>
Enter three sides of triangle:
3 4 5
Area of triangle = 6
Press any key to return to Quincy...

```

Figure 1.11

- **အဆို:** program မှာ ပေးခဲ့သည့် a, b, c s area တို့ကို identifier တွေကဲ့သို့ data type ကို double သို့မဟုတ် float ဝင်ပါသည်။
- Enter three sides of a triangle တို့ကို prompt ကို display မှာပို့ပြီး a, b, c ဆိုသည့်အတိုင်း (space ခြားပြီး) keyboard မှာပေးနိုင်ရမည်။ a, b, c တို့ကို ဆိုသည့်အတိုင်း $a=3, b=4$ နှင့် $c=5$ ကို ဝင်တတ်ရမည်ဆိုပါက။
- ခါတိုင်း မဟာမိတ် data မှာ ပေးခဲ့ခြင်းဖြစ်သည့် relational operator(>) နှင့် logical operator(&&) ရှစ်ခုကိုအသုံးပြုပြီး စစ်ဆေးပါသည်။ ဆိုလိုသကဲ့သို့ $(a = 3) + (b = 4)$ ဟု $(c = 5)$ ဟုဆိုသည့်အတိုင်း $(b = 4) - (c = 5)$ ဟု $(a = 3)$ ဟုဆိုသည့်အတိုင်း $(c = 5) + (a = 3)$ ဟု $(b = 4)$ ဟုဆိုသည့်အတိုင်း 5 နှင့် area တို့ကိုအကဲဖြတ်ပြီး Area of triangle = 5 သို့ display မှာပို့ ပေးနိုင်မည်။ sort() function မှာပေးရမည့်အတိုင်းမှာ စံနှုန်းတစ်ခုစီကို ဝင်ရမည့်အတိုင်း main function တွင်ပေးရမည်။
- ထပ်မံသိသည့်အတိုင်း data မှာပေးခဲ့သည့်အတိုင်း Triangle can't be formed ဆိုသည့်အတိုင်းမှာ စံနှုန်းတစ်ခုစီကို ပေးရမည်။ $a = 3, b = 4, c = 9$ ဆိုသည့် data မှာပေးခဲ့သည့်အတိုင်း run ဖြစ်ပြီး စစ်ဆေးပါသည်။ နှင့် (၁)၊ (၂) ကိုကြည့်ပါ။

```

bd Quincy 99
Enter three sides of triangle
12.5 20.8 18.5

Triangle can't be formed
Any key to return to Quincy...

```

(၃) (၁)၊ (၂)

Increment and Decrement Operators

C++ မှာ ဝါးတစ်ခု Unary operator ဆွဲကို သဘာဝ (၁, ၆) နားပါပြောပါသည်။ ဖြစ်ပေါ်လာသည့် ၅ operator ဆွဲကိုအသုံးပြုပြီး variable ဆွဲတစ်ခုကို တစ်ခုတည်းတစ်ခုတည်းလုပ်ဆောင်ပေးပါသည်။

ဇယား (၁.၆) **Unary Operators**

Symbol	Description
++	Increment operator
--	Decrement operator

၁။ ဖြစ်ပေါ်လာသည့် ပုံ (၁.၂၁) နားပါပြောသည့် Ex1010.cpp program ကို run ဖြစ်ပေါ်လာသည့် Unary operator ဆွဲတစ်ခုအသုံးပြုပြီး variable ဆွဲတစ်ခုကို တစ်ခုတည်းတစ်ခုတည်းလုပ်ဆောင်ပေးပါသည်။

```
Ex1010.cpp
// Listing 1.10. Using increment and decrement operators
#include <iostream>
int main()
{
    int kounter, oldCtr, newCtr;

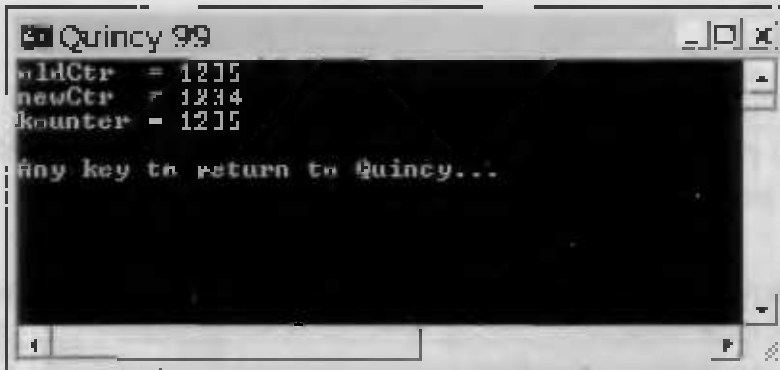
    oldCtr = 1234; // oldCtr = 1234
    newCtr = ++oldCtr; // newCtr = 1235, oldCtr = 1235
    kounter = newCtr--; // kounter = 1235, newCtr = 1234

    cout << "oldCtr = " << oldCtr << endl;
    cout << "newCtr = " << newCtr << endl;
    cout << "kounter = " << kounter << endl;
    return 0;
}
```

ပုံ (၁.၂၁)

program ကိုသေချာစာဖြည့်မယ်ဆိုင်

- ဆခွဲ၍ program မှာအသုံးပြုမည့် counter၊ oldCtr နဲ့ newCtr ဆိုတဲ့ identifier (3) မှာ ကို type int ဝင်အဖြစ်ပြုပါမယ်။ ဆနက်ရှင်ဟစ်မကြားရဘူး။ oldCtr ကို 1234 ဝင် assign ပေးပါမယ်။
- oldCtr ကန်ဖိုက်ကို increment ကန်ဖိုက်ပေးတဲ့ newCtr နဲ့ညီပေးပါမယ်။ ဒီတော့ newCtr = 1235 နဲ့ oldCtr = 1235 ဖြစ်သွားမိလိမ့်။
- ဒီတစ်ခါ counter ကန်ဖိုက်ကို newCtr = 1235 ဆိုတင် assign လုပ်ပြီး newCtr ကန်ဖိုက်ကို decrement ပေးပါမယ်။ ဒီတော့ newCtr = 1234 ဖြစ်သွားမိလိမ့်။
- ဆနက်ရှင်ဟစ် statement (3) မှာအသုံးပြုမည့် oldCtr၊ newCtr နဲ့ counter ကန်ဖိုက်ပေးတဲ့ ကန်ဖိုက်ကို display ပေးမိလိမ့်မယ်။ အဆင့်ကို ၆ (၁.၂) မှာဖြည့်ပါ။



```
Quincy 99
oldCtr = 1235
newCtr = 1234
counter = 1235

Any key to return to Quincy...
```

၆ (၁.၂)

Compound Assignment Operators

C++ မှာ ပါမိမိတဲ့ +=, -=, *=, /=, %=, <<=, >>= တို့ကို Compound assignment operator တို့ကို ခေါ်တယ်။ (၁.၇) မှာအသုံးပြုထားပါမယ်။ ၆ operators ပေးထားမိလိမ့်။ statement ဆွေးတဲ့ short form ဖြစ်တော့ အကြောင်းအရာလို့ပေးပါမယ်။ ပေးတာ sum = sum + counts; ကို sum += counts; လို့အရာလို့ပေးတယ်။

Symbol	Description
+	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
<<=	Shift left assignment
>>=	Shift right assignment
&=	Bitwise AND assignment
=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment

၁။ ငွ် (၁.၂၃) မှာဖော်ပြထားတဲ့ program ဟာဆိုရင် multiplication assignment operator ကိုအသုံးပြုပြီး အထောက်အကူပြုဆောင်ရွက်တဲ့ factorial ကိုရှာဖွေတဲ့ program ပါ။ ဆေးယာန်ကြည့်ပါ။

၂။ ဒီ program ကို run လိုက်မယ်ဆိုရင် ကွန်ပျူတာမှာ Enter a number : လို့ prompt ဟန်နာပေါ်လာစိမ့်မယ်။ ကွန်ပတ်တိုက် keyboard ဘေးထဲထဲမှာ 10 ကိုနှိပ်ပေးရင်ဆိုရင် ENTER key နှိပ်လိုက်တာနဲ့ 10 တန်ဖိုးကို ထွက်ပေးလိုက်ပါပြီ။ ငွ် (၁.၂၄) ကိုကြည့်ပါ။

၃။ Ex1011.cpp program ကိုအသုံးပြုအကြည့်မယ်ဆိုရင်

- ဟာရင် program မှာအသုံးပြုမယ့် identifiers i နဲ့ num ကို type int လို့အကြောင်းပြောပြီး fact ကို long int လို့အကြောင်းပြောပါကတော့ - တာ counter ပါ။ num က ကွန်ပတ်တိုက် factorial ရှာဖွေချက်ကိုကိုင်ဆောင်ပါ။ fact က num ကို store လုပ်မယ့် identifier ဖြစ်ပါကတော့။

```
Ex1011.cpp
// Listing 1.11: Using a compound operator

#include <iostream>
int main()
{
    int i, num;
    long int fact;

    cout << "\nEnter a number : ";
    cin >> num;
    fact = 1;
    for (i = 2; i <= num; i++)
        fact = i * fact;
    cout << "\nFactorial of " << num << " is " << fact << endl;

    return 0;
}
```

Figure 1.11

```
Quincy 99
Enter a number : 10
Factorial of 10 is 3628800
Any key to return to Quincy...
```

Figure 1.12

- `cout << "\nEnter a number : ";` ဆိုတဲ့ statement ကေး prompt ကိုရောင်းရောင်းကောင်းကောင်း keyboard ကနေ 10 ကိုထည့်သွင်းပါသည်။ ဒါ့ကြောင့် `num = 10` ထဲ `cin >> num;` ကနေ assign လုပ်သွင်းရင်ပဲ။


```

Ex1012.cpp
// Listing 10.2 Using the conditional operator

#include <iostream>
int main()
{
    float num1, num2, num3, largest;

    cout << "\nEnter three numbers:\n";
    cin >> num1 >> num2 >> num3;

    largest = num1 > num2 ? num1 : num2;
    largest = largest > num3 ? largest : num3;
    cout << "\nLargest number is " << largest << endl;
    return 0;
}

```

Figure 10.1

```

Quincy 99
Enter three numbers:
158.56
-588
2588

Largest number is: 2588

Any key to return to Quincy...

```

Figure 10.2

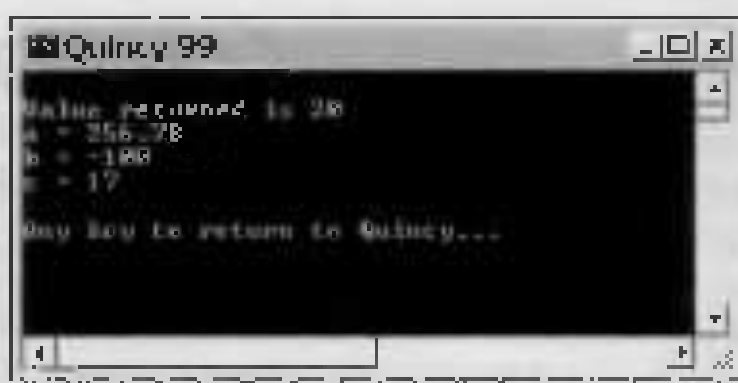
Ex1012.cpp program (Figure 10.1) is shown below.

- In this program, four identifiers (4) are of type float. They are num1, num2, num3, and largest. The data type of these identifiers is float.

program ဝယ်ရပုံ၊ comma operator သုံးစွဲနေပုံနှင့် output ဝင်ရိုး (3) ပုံရိပ်ကမ်းပေးဖို့ အောက်ဖြေထားတဲ့ program ကို ကြည့်ကြည့်ပါ။

Ex 11:3.cpp program ကို run လိုက်မယ်ဆိုရင် အောက်ဖြေထားတဲ့ (၁, ၂) အခြေအနေအထားပေါ်လာမှာပဲ။ ဒီ program ကို အောက်ဖြေတဲ့အတိုင်း ဆိုရမယ်။

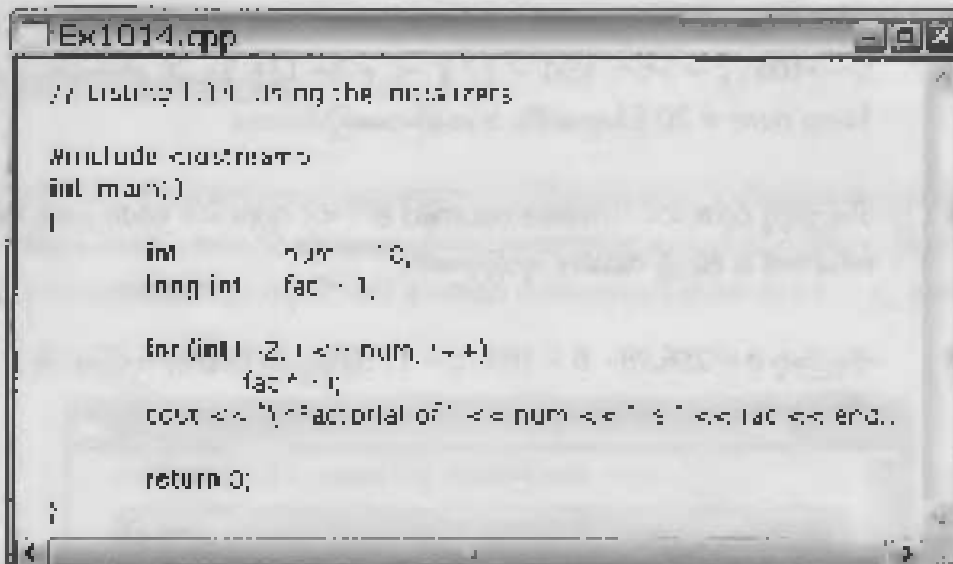
- ဝင်ရိုး program မှာ အောက်ဖြေတဲ့ identifiers (4) မှာ type float သုံးစွဲပြီးတဲ့အတွက် သုံးစွဲတဲ့ variable data သတ္တိကို $a = 255.78$, $b = -100$ နှင့် $c = 18$ ဆို၍ assign လုပ်ပေးမိတယ်။
- $num = (a++, b+3, --c-3)$ သုံးစွဲတဲ့ statement သွေး $a = 255.78 + 1 = 256.78$, $b = -100$, $c = --c - 18 - 1 = 17$ နှင့် $--c + 3 = 17 + 3 = 20$ ဆိုတဲ့အတွက် သွေးပါနား။ $num = 20$ [အောက်ဖြေတဲ့] ကို အောက်ဖြေထားတဲ့ [အောက်ဖြေတဲ့] ကို ပေးရမယ်။
- အောက်ဖြေတဲ့ `cout << "Value returned is " << num << endl`; သွေး Value returned is 20 ကို display လုပ်ပေးမိတယ်။
- အောက်ဖြေတဲ့ $a = 256.78$, $b = 100$, $c = 17$ ဆိုတဲ့အတွက် display လုပ်ပေးမိတဲ့ (၁, ၂) ကို အောက်ဖြေပါ။



ပုံ (၁၁-၃)

Initializers

ကျွန်တော်တို့ ရေးသားနေသော program မှာ variable ကို type declare လုပ်ပြီးမှ ထုတ်ပေး initial value ကို assign လုပ်နေခဲ့ပါသည်။ ၎င်းပင် type declaration မှာ data initialization ကို လုပ်နေထိုင်နေပါသည်။ ပုံ (၁-၂၅) မှာပင်ဖြစ်တဲ့ Ex1011.cpp program ကို ပုံ (၁-၂၆) ကလိုက်နာတတ်ပြီး ရေးပြီး run လိုက်မယ်ဆိုရင် ပုံ (၁-၂၆) ကထာဝရဖြစ်ရမည်။ ဂါန့်ရင် C++ မှာကတော့ အခြေခံအားဖြင့် ကိုယ်တို့ အနေဖြင့် နေရာပေးရမည့် အင်္ဂါအစား အိမ်ခြံမြေကို တင်ပါမည်။



```
Ex1014.cpp
// Listing 1.14 Using the initializers
#include <iostream>
int main()
{
    int    num = 10,
    long int fac = 1;

    for (int i = 2; i <= num; i++)
        fac *= i;
    cout << "\n factorial of " << num << " is " << fac << endl;

    return 0;
}
```

ပုံ (၁-၂၆)

ခေ့ဒ်သွေကို calling function ယခု ရှိသော နေရာမှနေ၍ ယာယီလွှဲ new function ဝေခံ၍ ဝေခံပေးရန် အင်္ဂါကိုသွေလိုက်ပြန်သို့ လုပ်ပိုင်ခွင့်ရနေပြီဟုလျှောက်ထား။ ပိုမိုခိုင် calling function သွေမှာ code မသွေပိုင်လွှဲလွှဲသော် C++ programming technique ဟုလည်း ခေ့ဒ်ခေ့ဒ်ဟုလျှောက်ပါ။

၂.၁ Create a Simple Function

၁။ function ဝေခံရာ အင်္ဂါပိုင်တဲ့အပိုင်း (2) နား function header နဲ့ statement body တို့ ပြန်ပါသည်။ function header ကိုလက်ခံစိတ်ကြိုညှစ်ရန် အပိုင်း (3) နုပါဝင်နေတာကိုတွေ့ရမှာပဲ။ (၁) return type (၂) function name နဲ့ (၃) parameter list တို့ပြန်ကြပါသည်။ return value ဟာ C++ data type သွေဝေ တစ်ခုခုဖြစ်နိုင်သလို data type သွေကို pointer ဘုန်းအော် pointer တစ်ခုလည်းဖြစ်နိုင်ပါသည်။ structure ၊ array သွေကို point ဘုန်းအော် pointer တို့လည်းဖြစ်နိုင်ပါသည်။ function name ဝေခံ identifier သွေကို နားပိုင်ပေးပေးနိုင်ရန်အတွက်ပါ။ parameter list ဟာ blank ဖြစ်နိုင်သလို variable သွေသွေကြိုခိုနေလို့လည်းပါသည်။ variable သွေဝေကို calling function က argument သွေကိုလက်ခံပေးပေးသော် statement body ကို program ဝေခံရာနေရာတိုင်းလျှောက်ပါသည်။ local variable declaration သွေ ၊ executable code သွေပါပိုင်ပါလိမ့်မည်။ ရှိနေပေက်တို့ function ဝေခံရာကို call ပေးခိုင် function parameter type သွေနှင့် return value အပိုင်းအစသွေဟာ အသွေပါပိုင် compiler ကို အခိုင် အသိပေးရပါမည်။ အင်္ဂါကို function declaration ကို prototype လို့ခေါ်ပါသည်။ အောက်ရှိ ၊ လွှဲသွေတဲ့ function ဝေခံရာ define လုပ်ပြီး main() function ဝေခံ call ခေါ်ကြည့်ပေးအောင်။ နဲ့ (၂.၁) ဝေ Ex201.cpp program လျှောက်ကြည့်ပါ။

၂။ Ex201.cpp program ကို run လိုက်သော်အခိုင် ကွန်ပူတာမှ င (၂.၂) နုပြသတဲ့ပေးပိုင်ပါပေတို့မှာပါ။ ဒီ program ကိုလျှောက်ကြည့်အပ်အိုရင်

- ခေ့ဒ်မှာ main() function လှည့်ပိုင်တို့မှာ func အညွှန်း functionprototype ကိုခေါ်ကြသော် return type က void ဖြစ်ဖို့ parameter list မှ variable တခုပေးပိုင် prototype declaration ခေ့ဒ်မှာ semicolon (;) နှိတ်ပေးရပါမည်။
- main() function ယခု ခေ့ဒ်တိုက်ခွဲဟော STEP 1: Inside the main() function ဆိုတဲ့စာမကြေးကို display လုပ်ပေးပါ။ ဒါဟာ နဲ့ (၂.၂) က လျှောက်ကြည့်ပါ။

```
Ex201.cpp
// Listing 2.1 Create a simple function

#include <iostream>
using namespace std; // declare the function prototype

int main()
{
    cout << "(STEP I) Inside the main() function!\n";

    // Call to the function
    func();
    cout << "(STEP II) Back in the main() function again!\n";

    return 0;
}

void func()
{
    // function body
    cout << "(STEP III) Now inside the func() function!\n";
}
```

④ ()

```
Quincy 99
STEP I: Inside the main() function
STEP II: Now inside the func() function
STEP III: Back in the main() function again
Any key to return to Quincy...
```

④ ()

Printing a Line of Asterisks

- `starline()` function (void) call `starline()`; program output
- asterisk(*) output
- `Ex202.cpp` program as code

```
Ex202.cpp
// Using 3.7 This program demonstrates a simple function
// that prints a line of 42 asterisks.

#include <iostream>
using namespace std;

int main()
{
    cout << "Data Type          Range" << endl;
    starline();
    cout << "char          ' 28 to 127'" << endl;
    cout << "int           -32,768 to 32,767" << endl;
    cout << "long          -2,147,483,648 to 2,147,483,647" << endl;
    starline();

    return 0;
}

void starline()
{
    for (int i=0; i<42; i++) cout << "*";
    cout << endl;
}
```

- `Ex202.cpp` program as run
- asterisk(*) output
- `Ex202.cpp` program as code

ခက်ကပ်ပုံ (2) မှာ Passing by value & Passing by reference ပုံနှစ်ခုတို့ကို ဖော်ပြထားပြီး (၂) မှာ
 Ex203.cpp program မှ constant value ကို pass လုပ်နိုင်ပုံကို ဖော်ပြထားပါသည်။

```

Ex203.cpp
// 1 - func.c - Calculate the area of a triangle

#include <iostream>
using namespace std;

int main()
{
    float area = calArea(10,20);
    cout << "Base = " << 10
    << " Height = " << 20
    << " The area of the triangle = " << area << endl
    << "\n";
}

float calArea(float b, float h)
{
    float area = 0.5*b*h;
    return area;
}
    
```

ပုံ (၂) (၆)

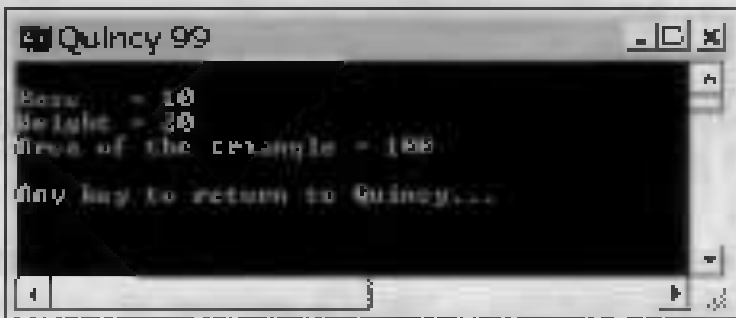
Ex203.cpp program ကို main ကိုယ်စားလှယ်နှင့် ဝန်ထုပ်ပေးပုံ (၂) မှာ ဖော်ပြထားပြီး အောက်ဖော်ပြပါအတိုင်း ဝန်ထုပ်ပေးပုံ နှင့် program ကိုယ်စားလှယ်တို့ကို

- program ကိုယ်စားလှယ် main() function မှာ calArea ဝန်ထုပ်ပေးပုံကို prototype ကို ဖော်ပြပြီး ဝန်ထုပ်ပေးပုံ return type ကို float ဖြစ်ပြီး parameter list မှာ float data(2) မှာကို pass လုပ်နိုင်ပုံကို ဖော်ပြပါသည်။
- main() function မှာ ဝန်ထုပ်ပေးပုံအတွက် identifier area ကို float ကို declare ပြုပြီး calArea() function ကို call လုပ်ပေးပုံကို ဝန်ထုပ်ပေးပုံ argument မှာ constant value ဝန်ထုပ်ပေးပုံ 10 ကို base ဖြစ်ပြီး 20 ကို height ကို ဝန်ထုပ်ပေးပုံ ဖော်ပြပါသည်။

- pass လုပ်ငန်းများကို constant argument (2) ၏ function calArea() ကဲ့သို့ B နှင့် F ၏ variable များသည်မှာ အတိအကျ၊ copy မည်သို့လုပ်ကိုင်သည်။ ဒီကဲ့သို့ B = 10 နှင့် F = 20 ဖြစ်သွားပါပြီ။ ဒီနေရာမှာ သတိထားရမှာက prototype declaration နှင့် function header ကိုသာတွင်ရပါမည်။ အခုလို data pass လုပ်ကိုင်တဲ့ passing by value လို့ခေါ်ကြသည်။

- အခုလိုရင် function body မှာလုပ်ကိုင်နေပါပြီ။ local variable ဖြစ်တဲ့ area နဲ့ data type ကို area လို့အမည်ပေးကြပြီး ကြိုတင်ရေးထားတဲ့ (base * height)/2 ၏အတိုင်း သင်္ချာပုံစံဖြင့် တွက်ပါသည်။ float area = 0.5 * b * h; ဆိုတဲ့ statement မှာ အဲဒီထဲတွင် parameter မှာ area ကိုတွက်ပေးရတာပဲ return area; ကိုပေး function return value မှာဖြင့် area ကို return လုပ်ပေးပါလိမ့်မည်။ ဒီကဲ့သို့ area နဲ့ data type နှင့် return value data type တို့ကတော့ပင် တူရမှာပေါ့။

- main() function ကိုဖြင့်ပေးပိုင်လာတဲ့အခါမှာ area = calArea(10,20) = return value လို့ assign လုပ်ပေးမှာဖြစ်ပါသည်။ main() function မှာ local variable မှာ area မှာ ကြိုတင်ရေးထားတဲ့ကိစ္စက စတင်ပေးပါပြီ။ ပေးပိုင်လာတာပေါ်မှာ cout << expression ကိုအသုံးပြုပြီး base height နှင့် area ကိုခွဲတစ်ခုကိုတစ်ခုတည်းမှာ display လုပ်ပေးလိက်ပြုနိုင်ပါသည်။ ပုံ (၂.၅) ကိုကြည့်ပါ။



ပုံ (၂.၅)

Ex204.cpp အားဆီရင် Ex203.cpp ကို အသစ်ပြင်ပေးထားတာပါ။ variable များကို pass လုပ်ရန် နှင့် program ကိုပြင်ပေးထားပါတယ်။ ပုံ (၂.၆) ကိုကြည့်ပါ။ ဒီ program ကိုလေ့လာကြည့်မယ်ဆိုရင်

initialise statement for base, height, area. 2) global variable and declare and prototype (3) and define (4) main() and call (5) and call (6) and call (7) function (8) call and declare (9) function and base & height and (10) and (11) and (12)

```

Es204.cpp

// Listing 2.6: Pass of variables by value
#include <iostream>

float base, height, area;
void getData();
void calcArea();
void printArea();

int main()
{
    getData();
    calcArea();
    printArea();

    return 0;
}

void getData()
{
    cout << "Enter base and height:\n";
    cin >> base >> height;
}

void calcArea()
{
    area = 0.5 * base * height;
}

void printArea()
{
    cout << "Area of the triangle is: " << area << endl;
}

```

Ex4. main() ဝင်ရောက်ပေးသောအခါ callArea() function ကို call ပြန်ပေးပြီး callArea() function ထဲမှပတ်သက်သောအခါတွင် ဤကိန်းများကို လိုက်နာရမည်ဖြစ်သည်။ base နှင့် height များကိုပေးသည့် function ကိုဖော်ပြသောအခါတွင် area တွက်ပြီးနောက် main() ကိုပြန်ပတ်သက်သည့် printArea() ကို call ပြန်ပေးပြီးနောက် printArea() function ကို area ကိန်းကို print လုပ်ပြန်ပေးသည်။

Ex204.cpp program ကို run လုပ်ပါသည်။ အောက်ဖော်ပြပါပုံ (၂-၉) မှာရောက်သောအခါ၌ main() ကိုဖော်ပြသော Ex203.cpp program ကိုဆက်လက်



ပုံ (၂-၉)

Creating a Self-defined Square Root Function

Ex5. အောက်ဖော်ပြပါ program တွင် sqrt() ကို 10 ကို self-defined function တစ်ခုကို ဖန်တီးပြီး ဖော်ပြသောပုံ နှစ်ပတ်လုံးပုံဆိုင်ရာများရှိ linear ဖော်ပြချက်များ calling function ဝင်၊ constant ၁၀ ကို called function သို့ pass ပြန်ပေးသည့်အခါ၌ အိမ်မှာထွက်လာပုံဆိုင်ရာများကိုဖော်ပြသော အခါ၌ ကိုယ်တိုင် ဖော်ပြချက်များကို main() ကိုပြန်ပေးသည့် အခါ၌ ဖော်ပြသော Ex205.cpp program ကိုဆက်လက်ပြန်ပါသည်။ ပုံ (၂-၁၀) မှာဖော်ပြသော Ex205.cpp program ကိုဆက်လက် ဖော်ပြပါသည်။

```

Ex205.cpp
// Listing 2.5 This program displays the square root of
// a number passed from a program to the function.

#include <iostream>
void sqrtot (float , float );

int main()
{
    sqrtot (625, 10000);

    return 0;
}

void sqrtot (float  xn, float  xr)
{
    for (int i=1; i<=50; ++i)
        xr = (xr+xn)/2;
    cout << "Root of " << xn << " is " << xr << endl;
}

```

ပုံ (၂. ၁၁)

Ex205.cpp program ကိုမေးခွန်းထုတ်ဖော်ချက်

- program မေးခွန်း main() function ရဲ့အပေါ်မှာ sqrtot နာမည်နဲ့ prototype ကိုကြိုတင်ပေးထားပြီး return type က void ဖြစ်ပြီး parameter list မှာ float data(2) မှာနဲ့ pass လုပ်နိုင်မှာပါ။ မေးခွန်း main() function မှာ sqrtot (625, 10000) ကို call မှုပြီး 625 နဲ့ 10000 ကို sqrtot() function ကို pass လုပ်ပေးပါသည်။ pass ကို by value နည်းပေးပါ။
- sqrtot() function header မှာ xn = 625 နဲ့ xr = 10000 ဆိုပြီး constant ပေး pass လုပ်ပေးထားပြီး function body ထဲမှာ xn ကို square root ကိုရှာဖွေပါသည်။ xn ကို ထွက်ကုန်ကို assume လုပ်တဲ့ root ကိုယူပြီး root ရှာတဲ့အခါမှာ xr = (xr+xn)/2 ဆိုတဲ့ statement ကိုထပ်ထပ်ဆက်ဆက်ပြုလုပ်နေရမည်။ for loop ကိုလုပ်ပြီး (50) ပတ်လုံးလုပ်ဆောင်ပြီး output result ကို ပုံ (၂. ၁၁) မှာပေးပြောပါသည်။



Figure 2.5

Figure 2.5 program `sqrt.c` is a simple program that demonstrates the use of a function. The program prompts the user for a number and then displays the square root of the number. The program is shown in Figure 2.5.

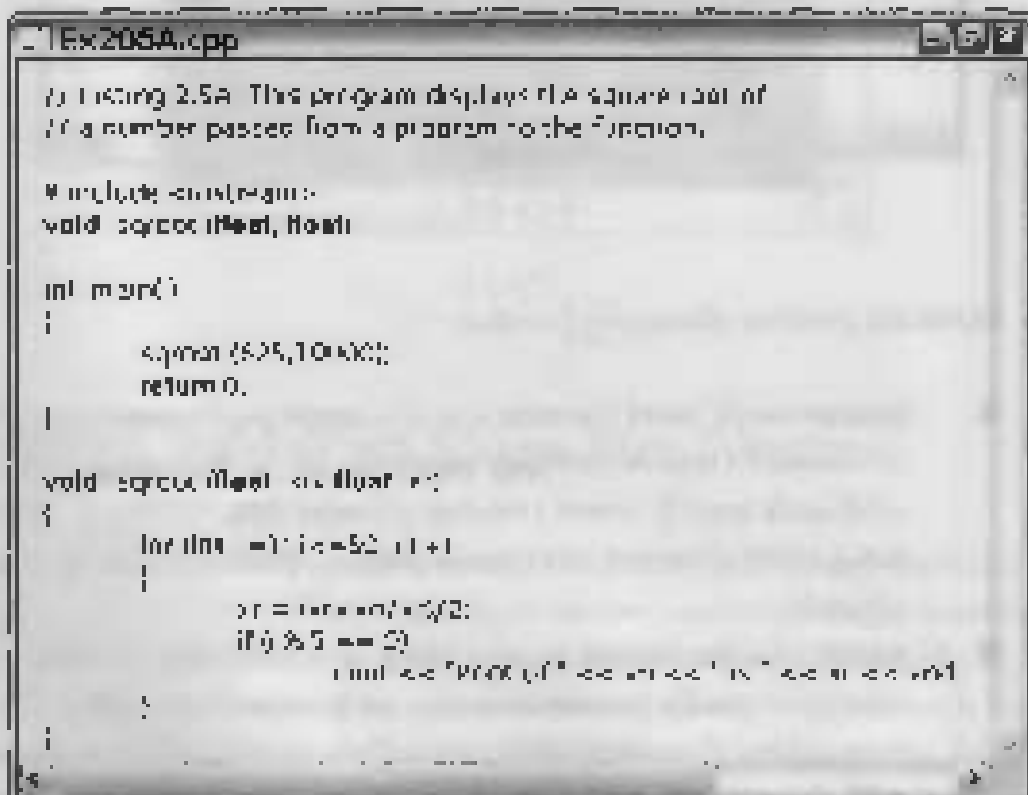


Figure 2.6

၅) နှစ်ခြားကိန်းများ သတ်မှတ်ပေးကာ for loop ထိုးထွင်း (50) ကိန်းတွင်ပိုင်းခြား ခန့်မှန်းချက်များကို ပြန်လည်ပြုလုပ်ပါ။ ဘယ်ကြိမ်မျိုးကို မှတ်တမ်းတင်မည်ဆိုသော်လည်း (၂၅ ခု) ကန့်သတ် display ပြုလုပ်ပေးခြင်းပါသည်။ စာရေးဆရာပုံကိစ္စ program trace ဆိုက်ကြည့်ပါ။



ပုံ (၂. ၁၃)

၆) ၆) program မှာ root ရာခိုင်နှုန်းကို assumed value ထိုးထွင်း keyboard ထဲမှ ဖတ်ယူခြင်းမရှိပဲ ပါဝင်သော ၆) program ကိုကူးယူပါ။ x၊ y နှင့် x၊ y ထိုးထွင်းချက်ကို variable ထဲမှထုတ်ပေးပေးရန် ပုံစံပြင်ဆင်ပေးပါ။ (၂. ၁၄) မှတစ်ဖက်က program code ဖတ်၍ အားပြုခြင်းများအားလုံးကို သတိပြု ပြုလုပ်ပေးပါ။

- sqrt(x) ; function မှ return type ကို float ထိုးထွင်းပေးထားသည်။ main() ထဲမှ x၊ y ကို ဖတ်ယူချက်ကို keyboard ထဲမှ ဖတ်ယူခြင်းမရှိပဲ ပါဝင်သော ၆) program ကိုကူးယူပါ။ x၊ y နှင့် x၊ y ထိုးထွင်းချက်ကို variable ထဲမှထုတ်ပေးပေးရန် ပုံစံပြင်ဆင်ပေးပါ။
- sqrt(x) ; function ထဲမှထုတ်ပေးရန် ပုံစံပြင်ဆင်ပေးရန် ပုံစံပြင်ဆင်ပေးပါ။ sqrt(x) မှ return ထုတ်ပေးပါ။

၆) ၆) program ကို run ထုတ်ပေးခြင်းကို (၂. ၁၅) ပုံစံပြင်ဆင်ပေးခြင်း ပြုလုပ်ပေးပါ။

```
Ex2058.cpp

// Listing 2.58. This program displays the square root of
// a number passed from a program to the function.

#include <iostream>
float sqrt(float);

int main()
{
    float x, xr;

    cout << "Enter a positive real number:\n";
    cin >> x;
    cout << "Assume its root:\n";
    cin >> xr;
    cout << "Root of " << x << " is " << sqrt(x) << endl;
    return 0;
}

float sqrt(float x, float xr)
{
    for (int i=1; i<=50; i++)
        xr = (xr + x/xr)/2;
    return xr;
}
```

Figure 2.58



Figure 2.59

Pass Variables by Reference

၁၈ C++ program သို့မဟုတ် calling function ထဲမှ argument တစ်ခုကို called function ထဲမှ သေးသည့်အခါမှ argument-to-argument ခြေထောက်ပုံစံဟု ခေါ်ဆိုပြီး calling argument ပြုပြင်ပေးပေးရာမှ address မှ ခြေထောက်ပုံစံဟုခေါ်ဆိုပြီး argument-to-address pass သုံးခုကို Passing by Reference သုံးခု ခေါ်သည်။ Ex203.cpp program ကို ရှိပုံအတိုင်းပြုပြင်ပေးထားပြီး (၂. ၁၆) နှင့်မတူဘဲ Ex205.cpp program ကိုရေးပေးကြည့်ရအောင်။

```

Ex205.cpp
// Listing 2.6: Passing variables by reference

#include <iostream>
using namespace std;
void printArea(float x);

int main()
{
    float base, height, area;

    cout <<< "\nEnter base and height:\n";
    cin >> base >> height;
    calcArea(base, height, area);
    printArea(area);
    return 0;
}

void calcArea(float b, float h, float &x)
{
    x = 0.5 * b * h;
}

void printArea(float x)
{
    cout <<< "\nArea of the triangle is " << x << endl;
}

```

(၂. ၁၆)


```
EX207.cbp
// Listing 2.7 This program reads a lowercase character,
// converts it to uppercase and then writes out
// the uppercase equivalent.

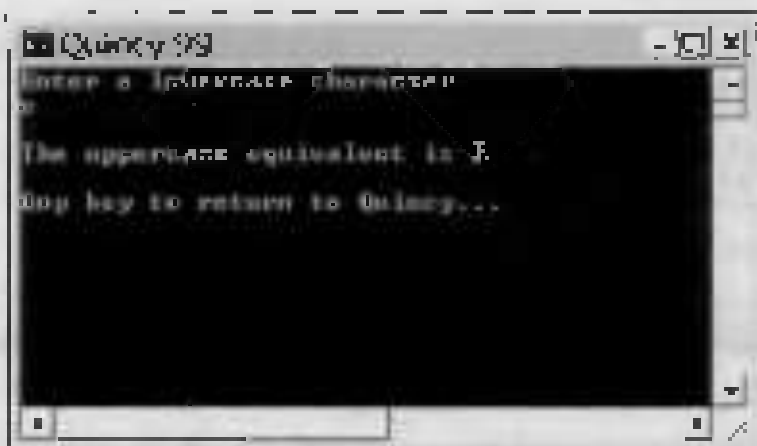
#include <stdio.h>

void lower_to_upper (char c1, char& c2)
{
    c2 = (char) (c1 + 26 - (c1 >= 'a' ? 'a' - 'a') : c1);
}

int main()
{
    char lower, upper;

    printf ("Enter a lowercase character: ");
    while (lower)
        lower = getc(stdin);
    lower_to_upper(lower, upper);
    printf ("%c is the uppercase equivalent of %c\n", upper, lower);
    return 0;
}
```

() ()



() ()



ပုံ (၂.၂၀)

Default Function Arguments

function prototype တစ်ခုကို declare လုပ်တဲ့အခါမှာ parameter list မှာ ပါဝင်တဲ့ argument များထဲမှာ ပြန်လည်အသုံးပြုရတဲ့ default value ဆောင်ရွက်ထားနိုင်ပါသည်။ function call မှာ လိုအပ်တဲ့ parameter argument မတွေကိုမရဘဲ compiler ကနေ default value ဆောင်ရွက်ပေးပေးပါ။ ပုံ (၂.၂၁) ယခုပုံအတိုင်း program ယူဆီရင် function prototype မှာ argument မတွေအတွက် default value ဆောင်ရွက် declare လုပ်ပြီး အသုံးပြုခွင့်ပေးတဲ့ အတိုင်းအတာပါ။

ဒီ program ကို လိုက်နာကြည့်ပါအံ့။

- မူရင်း function prototype volume() ကို declare လုပ်တဲ့အခါမှာ argument မတွေအတွက် default value မတွေကို 2, 3, 4 ကို အသုံးပြု ဆက်သွယ်ပေးပါသည်။ အထူးသဖြင့် call မှုကို volume() function မှာ argument မရှိပေမယ့်အတွက် default value (3) ကို လိုက်နာဆောင်ရွက်ပြီး volume ကို $1 * w * h = 2 * 3 * 4 = 24$ ကို တွက်ပေးပါသည်။
- ခုတ်ယူဆောင်ရွက် call မှုကိုအခါမှာ parameter list မှာ argument တစ်ခုပဲ ပါဝင်တဲ့အတွက် အထူးသဖြင့် default value ကို new argument နဲ့ အသုံးပြုပေးပေးပါ။ ဒီအတိုင်း volume = $5 * 3 * 4 = 60$ ဖြစ်ပေါ်ပါ။ ဒီနည်းအတိုင်းအတိုင်း function မတွေကို call မှုကိုမူရင်း volume ကို $5 * 7 * 4 = 140$ နဲ့ $5 * 7 * 9 = 315$ ကို ဖြစ်ပေါ်စေရမည်။ ပုံ (၂.၂၂) မှာ Ex209.cpp ကို run ပြုလုပ်ပါသည်။ လိုက်နာကြည့်ပါ။

```
Ex209.cpp
// Listing 2.9 Using functions with default arguments
#include <iostream>
using namespace std;
void volume(float=2, float=3, float=4)
{
    int i=0;
    while (i<3)
    {
        volume(i) // all three arguments default
        volume(i, 5) // provides the first argument
        // provide the first and second arguments
        // volume(5, 7)
        volume(5, 7, 9) // provide all three arguments
        return 0;
    }
}
void volume(float f, float w, float h)
{
    float vol = f*w*h;
    cout << "Volume = " << vol << endl;
}
int main()
{
    volume(5);
    volume(5, 7);
    volume(5, 7, 9);
    return 0;
}
```

Figure 2.10

```
Quincy 99
Volume = 24
Volume = 60
Volume = 140
Volume = 315
Try key to return to Quincy...
```

Figure 2.11

Inline Functions

အောက်ဖော်ပြပါ function တစ်ခုကို define လုပ်လိုက်တာနဲ့ ကွန်ပိုဆီဇာ compiler ကနေ ဘယ် function ကို execute လုပ်မှာလဲ၊ instruction ဘယ်တစ်ခုကို create လုပ်မလဲ၊ memory မှာ ညီလာခံတိုက်ပါလို့၊ function ကို call မှုကိုမေး calling function ကနေ called function ကိုကို ပြန်မကုတ်တော့ဘူး။ ပျက်စီးနေတာလား။ ငါ့ဆီက ငါ့ဆီက calling function ကိုပြန်လာမယ်လို့ function ကို (19) ခြိမ် call မှုကို (10) ခြိမ်တိုက် ပြန်မကုတ်တော့ဘူး။ ဘာလို့လဲ။ ဘာလို့လဲ။ call မှုကို function က ညီလာခံတိုက်တာဆိုရင် execution time ကနေ၊ traverse time ကနေ၊ အဲဒါနဲ့ function ကလည်းကောင်း၊ call မှုကိုကလည်းကောင်း၊ inline function ကိုမေးခြင်း၊ call မှုကိုမေးခြင်းနဲ့ inline function တစ်ခုကို create လုပ်မှာမှာ keyword inline ကို function declaration လုပ်မှာမှာတည့်မတည့်မဟုတ်ပါ။ Inline function ကိုမေးခြင်းကနေ compiler က ညီလာခံတိုက် function ကို create လုပ်မလားလို့၊ function call မှုကိုမေးခြင်း၊ inline function မှုကိုမေးခြင်း၊ code statement တွေကို calling function မှု ကိုက်စိုက်တာကိုမေးခြင်း၊ ကာလကိုမေးတာကိုမေးခြင်း၊ လိုက်နာ။ ဒါတွေ inline function မှု code statement ဘယ်တုန်းကလည်းကောင်း၊ ပေးတာနဲ့ တည့်မတည့်မဟုတ်ပါ။ program ကိုမေးခြင်းဖြင့် inline function တွေက ပေး function တွေကို call မှုကိုမေးပါလား။

၂.၆ Local and Global Variables

local variables ဆိုတာ main() function နဲ့ တစ်ခုခု function ကို body ကိုမေး၊ declare လုပ်တာ၊ variable name ဆိုတာ၊ ဘာလို့လဲ။ ပုံစံအတိုင်းပဲ function တစ်ခုခုကိုမေးခြင်းပဲလား။ function တစ်ခုခုက variable တစ်ခုခုက ဆိုတာ function ကို variable ပုံစံအတိုင်းမေးတာပဲ။ program မှု တစ်ခုခုကိုမေးတာဆိုရင်မေးတာ မှု scope ကို မေးတာနဲ့ function ကိုမေးတာကလေး ဆိုတာကိုမေးတာ။ function header ဆိုတာ parameter list ဆိုတာကို မေးတာ function ကလေးက local variable တွေကိုမေးတာနဲ့ variable ဆိုတာ function body ကိုမေးတာနဲ့ declare လုပ်တာနဲ့မေးတာ။ global variable ဆိုတာ main() function ကိုမေးတာ၊ program ကိုမေးတာ၊ declare လုပ်တာနဲ့ variable scope ဆိုတာ၊ program တစ်ခုခုကိုမေးတာပဲ။ ပေးတာနဲ့ (၂.၅) မှု မေးတာနဲ့ C++2010 ခု program ကိုမေးတာနဲ့မေးတာ။

```

Cx2010.cpp

// Listing 2.10 This program creates a function power()
// to compute the value of integer raised to the yth power.

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    long int x,y,temp;

    cout << "Enter x and y:"
    cin >> x >> y;
    cout << endl;
    temp = power(x,y); // functional
    cout << x << " to the power of "
        << y << " = " << temp;
    cout << endl << "Using math function pow() "
        << pow(x,y) << endl;
    cout << endl;
    return 0;
}

long int power(int a, int b) // function header
{
    long int temp = 1;
    for (; b > 0; b--)
        temp *= a;

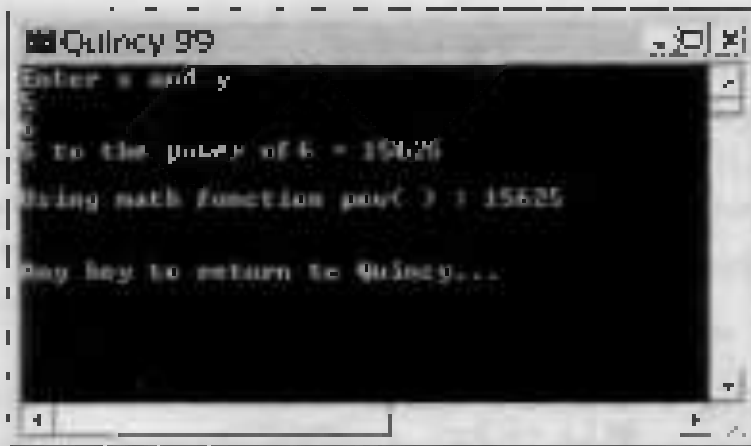
    return temp;
}

```

Fig 2.10 C++ program to compute the power of a number using a function.

- temp ဆိုတဲ့ variable name ကို main function မှာပဲ power() function မှာ (2) မှာပဲ သုံးစွဲနေတာကို သတိပြုရမယ်။ အခုဆိုတော့ main() function ကိုယ်တိုင်မှာပဲ main() function မှာပဲ long int ဆိုတဲ့ variable ကို သုံးစွဲနေတာကို သတိပြုရမယ်။

- x နှင့် y သတင်း data သွင်းထည့်ပေးပြီး power() function ကို call ခဲ့ သောအခါ
calling argument သဘာဝ x နှင့် y ဖြစ်ပါသည်။ power() function ကို မေးမြန်းသည့်အခါ
 x နှင့် y ကို ရိုက်ထည့်သွင်းတဲ့ function ကို main() က local variable သတ္တိရဲ့ a နှင့် b ပုံစံထဲ
ထည့်သွင်းထားပါသည်။
- power() function body ထဲက temp ကို main() function ထဲက temp ရဲ့အတိုင်း
ပေးသော အမှန်အတိန်ပါ။ function ကိုသွင်းထည့်ပြီး temp ကို return ခုတ်ပေး
သောက power() function ရဲ့အတိုင်း return ခုတ်ပေးသောအတိုင်း main() ကို
ပြန်သွင်းသည့်အခါ local variable ဖြစ်တဲ့ temp သည် power() function ရဲ့ value ကို
သွင်း temp ► power(a,y) ► 16384 ဖြစ်လာခြင်းဖြစ်ပါသည်။
- မေးခွန်းမှာ အဖြေ 12 နှင့်ပုံနှိပ်သည့်အတိုင်းပါသည်။ မတူသော အဖြေအတိုင်း ပြောပေး
သဖြင့် အတိုင်းပေးထားခြင်း မှ မှားမည်ဖြစ်သော main library function ဖြစ်တဲ့ power()
function ကိုသွင်းဖြည့် သွင်းထည့်သွင်းပါ။ ဝါလောက်အောင် local variable သွင်းထည့်ပြီး
အတိုင်းပေးခြင်းဖြစ်ပါသည်။ (၂.၂၅) မှာ Ex2010.cpp ကို run ဖြစ်ပါသည်။



(၂.၂၅)

- Ex2010.cpp ကိုနောက်ပိုင်းဆုံးဖြတ်ချက်ပါ။ (၂.၂၅) မှာပေးတဲ့ Ex2010A.cpp
program ကို global variable ကိုသွင်းဖြည့်ပါသည်။ Ex2010A.cpp သဘာဝ
program ရဲ့အတိုင်းပေးသော အတိုင်းပေးသည့် အတိုင်းပေးသောအတိုင်း temp ကို
global variable အတိုင်းပေးထားသည့်အတိုင်း main() ထဲက temp နှင့် power() ထဲက
temp ကိုပေးပါသည့်အတိုင်း variable ဖြစ်သည့် function body ကိုသွင်းထည့်ခြင်း

temp value ကို return လုပ်ပေးလိုက်တဲ့ main() ကို ပြန်ပေးတဲ့အတွက် အတိတ်ကဲ့သို့ပင် ဖြစ်နိုင်ပါသည်။ ဤကဲ့သို့ ဖြစ်ပေါ်ခြင်းသည် global variable တွေသာ function တွေဖြား သွားလာပြုမိတဲ့ data transfer လုပ်ပေးနိုင်တဲ့ အတွက် ရှိတဲ့အတွက်ပင်။ ပုံစံမှာ program (2) မှာကဲ့သို့ပင် ဖြစ်ပေါ်ကြမည်အတွက်ပင်။ program ကို run ပြုလိုက်ရင် အောက်ဖော်ပြပါအတိုင်း

```

Ex2010A.cpp
// Listing 2.10A. This program demonstrates
// the use of global variables.

#include <iostream>
void power (int, int);
long int temp; // define global variable

int main()
{
    long int x, y;

    cout << "Enter x and y\n";
    cin >> x >> y;
    cout << "Enter (C)\n";
    power (x,y);
    cout << x << " to the power of " << y << " = "
        << temp << endl;
    return 0;
}

void power (int a, int b)
{
    temp = 1;
    for ( int i = 0; i < b; i++) temp *= a;
}
    
```

(၂ - ၅)

Overloaded Functions

C++ program တစ်ခုမှာ function သုံးခုရှိပြီး function name တစ်ခုတည်းပေးပြီး create လုပ် ထားပါသည်။ မှတ်တမ်း function overload ကို လိုက်နာပါသည်။ function name တူသော parameter list ကွဲပြားစေရမည်။ argument type argument number ကွဲပြားစေရမည်။ ဥပမာ C library မှ `abs()` absolute value ရှင်းတဲ့ function သုံးမျိုးကောက်နိုင်ပါသည်။ integer ကတွက်ဆိုရင် `abs()`၊ long integer ကတွက်ဆိုရင် `labs()`၊ floating-point value ကတွက်ဆိုရင် `fabs()` ခန့်မှန်းထားပါသည်။ ဒီဗျာ မှတ်တမ်းကွဲပြားစေရမည်။ ကွဲပြားစေရမည်။ ဒီအခန်းကဏ္ဍကို C++ မှာ function overloading နည်း အသုံး ပြုနိုင်ခြင်းဖြစ်ပါသည်။ function overloading ကို function polymorphism လို့လည်းခေါ်ပါသည်။ poly မှာ many၊ morph က form ဝါ၊ မြားပြားခြင်းပုံစံအမျိုးမျိုးနဲ့) function တစ်ခုကို polymorphic function လို့ခေါ်လာပုံဖြစ်ပါသည်။

၅ (၂) (၆) မှာရှိသော Ex2011.cpp program ကိုလေ့လာကြည့်ရပါမည်။

- program များမှာ PI = 3.141593 နဲ့ radius၊ height၊ volume တို့ကို global float variable ကွဲပြားစေရမည်။ `calVol()` ကို overloaded function ကွဲပြားစေရမည်။ `main()` function ထဲမှာတော့ radius နဲ့ height တို့မှာပေးတဲ့ data ကွဲပြားစေရမည်။ keyboard ကနေရက်စာညှိရမည်။ ငြိမ်းတာရင် ထား `calVol(radius)` ကို call မှာပြီး `void calVol(float r)` function ထဲမှာ $volume = 4 * PI * r^3 / 3$ ဆိုတဲ့ formula မှာရက်စာညှိပြီး sphere volume ကိုတွက်ပေးရမည်။ volume ကို global လုပ်ထားတာနဲ့ return ပေးရမည်။ `main()` မှာ volume ကန့်သတ်တဲ့ အသုံးပြုမှုနဲ့ပေးပါ။
- ရောက်တင်ခါ call မှာတဲ့ `calVol()` function မှာ argument (2) ခုပါဝင်သည်။ radius နဲ့ height ကို argument မှာရက်စာညှိပြီး pass လုပ်ပေးရမည်။ `calVol(float r, float h)` ထဲမှာပေးပြီး $volume = PI * r^2 * h$ ဆိုတဲ့ formula မှာရက်စာညှိပြီး cylinder volume ကိုတွက်ပေးရမည်။
- program ကို run လိုက်ရမည်ဆိုရင် ၅ (၂) (၇) မှာရှိသော အသုံးပြုမှုနဲ့ပေးရမည်။ radius = 5.5 နဲ့ height = 7.5 ကိုတွက်ပေးကြည့်ပါ။ sphere နဲ့ cylinder volume ကွဲပြားစေရမည်။

```
Ex2011.cpp
// Listing 2.11 Using overloaded functions
#include <iostream>
const float PI = 3.141593;
float radius, height, volume;
void caVol(float);
void caVol(float, float);

int main()
{
    cout << "Enter radius : ";
    cin >> radius;
    cout << "Enter height : ";
    cin >> height;

    caVol(radius);
    cout << "\nVolume of sphere is : " << volume << endl;

    caVol(radius, height);
    cout << "\nVolume of cylinder is : " << volume << endl;
    return 0;
}

void caVol(float r)
{
    volume = 4/3*PI*r*r*r;
}

void caVol(float r, float h)
{
    volume = PI*r*r*h;
}

```

(U-3)

```
Quincy 99
Enter radius : 5.5
Enter height : 7.5

Volume of sphere is 696.91
Volume of cylinder is 712.749
Any key to return to Quincy...
```

(U-3)

Overloading Function myDate()

၁၀ နံ (၂.၂၅) နားပြားတဲ့ Ex2012.cpp program သေဆိုင်ရင် string type date နဲ့ integer type date တစ်နဲ့ (2) နှိုက်ရင် overloaded myDate() function သုံးခုကို သုံးခုလုံး display လုပ်နိုင်တယ်။
call ဆောင်ရွက်ရင်တော့ ကောင်းတယ်။

```
Ex2012.cpp
// Listing 2.12 This program shows function
// myDate( ) overloading

#include <iostream>
void myDate (char *str)      // date as string
void myDate(int, int, int)  // date as numbers

int main()
{
    myDate("20th Apr 2003",
    myDate(4,20,2003);
    return 0;
}

void myDate (char *str)
{
    cout << "Date : " << str << endl;
}

void myDate (int month, int day, int year)
{
    cout << "Date : " << month << "/" << day << "/" <<
    << year << endl;
}
}
```

နံ (၂.၂၅)

၂ ဒီ program ကို run လိုက်မယ်ဆိုရင် နံ (၂.၂၅) နားပြားတဲ့အတိုင်း Date: 20th Apr 2003 နဲ့ Date: 4/20/2003 ဆိုတဲ့အတိုင်း (2) နှိုက်ရင် display လုပ်ပေးမယ်။

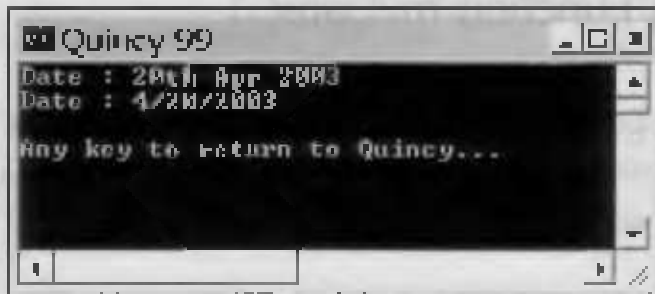


Figure 2.19

Overloading with Default Arguments

Ex2013.cpp program တွင် လုပ်ငန်းဆောင်ရွက်မှုများကို လုပ်ဆောင်နိုင်စေရန် repeat(char c) function prototype ထဲသို့ default argument ဆွဲဆောင်ပြီး overloaded လုပ်ငန်းဆောင်ရွက်မှုများကို function name တစ်ခုနှင့် function definition တစ်ခုတည်းအဖြစ် ပြောင်းလဲရေးရာကို ပြသထားပါသည်။ Figure 2.20 ကို ကြည့်ပါ။

```

Ex2013.cpp
// Listing 2.13 This program demonstrates function
// overloading with default arguments.

#include <iostream>
void repeat_ch(char c, int n = 10);

int main()
{
    repeat_ch('A');
    repeat_ch('B');
    repeat_ch('C', 10);
    return 0;
}

void repeat_ch(char ch, int n)
{
    for (int i = 0; i < n; i++) cout << ch;
    cout << endl;
}

```

Figure 2.20


```

Quincy 99
Enter a number : 100
Factorial of 100 is 9.33262e+157
Any key to return to Quincy...

```

ပုံ (၂.၈၆)

The Tower of Hanoi

၁) Ex2015.cpp program သည် recursive program အဖြစ်ပါယ်သော အယ်ဒီတာ့အပေါ်တွင် ဖန်တီးထားသော အယ်ဒီတာ့အတွက် ဘီနီ (3) တိုင်မှီပစ္စည်းအား အယ်ဒီတာ့အတွက် အသုံးပြုမှုကို အကြီးအပြားပုံစံဖြင့် အတိုးပေးအပ်ပေးရန် အမည်ကြီးသည့် မှတ်တမ်းတင်ပေးမှုပေးပါမည့်အပြင် program ကွန်ပိုက်လ်ရေးပါမည့် အားတိုင်ကို LEFT၊ ဒုတိယတိုင်ကို CENTER & တတိယတိုင်ကို RIGHT ခြေခံပါသော အမှန် disk (3) မှတ်တမ်းတင်ပေးရန် ပုံ (၂.၈၇) မှ ပြုစုထားပါသော အယ်ဒီတာ့အပေါ်ပါသည်။

```

Quincy 99
Welcome to the TOWERS OF HANOI
How many disks ? 3
Move disk 1 from L to R
Move disk 2 from L to C
Move disk 1 from R to C
Move disk 3 from L to R
Move disk 1 from C to L
Move disk 2 from C to R
Move disk 1 from L to R
Any key to return to Quincy...

```

ပုံ (၂.၈၇)

၂။ * program ၏ source code သူတို့ ၏ (.c, .cpp) နာမည်ပြောင်းကွင်း၊ ဆောင်းချက်တို့ကို တစ်ဆင့်တစ်ဆင့် အသုံးပြုနိုင်သည်။ မိမိအသုံးပြုသည့်အတိုင်း program သစ်ကိုလည်းကောင်း၊ ပြောင်းသစ်

```

Ex2015.cpp
// Listing 20.15. This program is a game used to transfer
// the disks from the leftmost pole to the rightmost pole,
// without ever placing a larger disk on top of a smaller
// disk. Only one disk may be moved at a time.

#include <iostream>
void transfer(int, char, char, char);

int main()
{
    int nDisk;

    cout << "Welcome to the TOWERS OF HANOI\n";
    cout << "How many disks? ";
    int n;
    cin >> nDisk;
    cout << endl;
    transfer(nDisk, 'L', 'R', 'C');
    return 0;
}

void transfer(int n, char from, char to, char temp)
{
    if (n > 0)
    {
        transfer(n-1, from, temp, to);
        cout << "Move disk " << n << " from " << from
            << " to " << to << endl;
        transfer(n-1, temp, to, from);
    }
}

```

၆ (၂ - ၃၆)

Passing Structure Variables

function သည် constant ဟု variable သတ်မှတ်ရန် pass ရန်လိုအပ်သည့် data type ဟု ပေး
အသုံးပြုသည့် structure သုံးမျိုးပေးသည့် pass ရန်ပေးသည့်နည်းကို ဝ (၂, ၃၆) ဖော်ပြထားသည်။ Ex2016.cpp
program သည် structure ဟု ပေးသည့် function သုံးခုကို အသုံးပြု၍ pass ရန်ပေးနိုင်ပါသည်။ ဟု ပေးသည်။

```
Ex2016.cpp
// Listing 2.16. This program demonstrates
// passing structure as argument.

#include <iostream>

struct height
{
    int    feet,
    float inches;
};

void convert(height h)
{
    cout << h.feet << " feet"
         << h.inches << " inches\n";
}

int main()
{
    height myHeight;

    cout << "Enter feet: ";
    cin >> myHeight.feet;
    cout << "Enter inches: ";
    cin >> myHeight.inches;
    cout << "My height is ";
    convert(myHeight);
    return 0;
}
```

၇ (၂, ၃၆)


```

// Listing 2.17: This program demonstrates returning a structure
#include <iostream>

struct Length
{
    int feet;
    float inches;
};

void convr(Length),
Length add(Length, Length);

int main( )
{
    Length piece1,piece2,total;

    cout << "For piece1\n";
    cin >> piece1.feet;
    cout << "\nEnter inches : ";
    cin >> piece1.inches;

    cout << endl;
    cout << "For piece2\n";
    cin >> piece2.feet;
    cout << "\nEnter inches : ";
    cin >> piece2.inches;

    total = add(piece1, piece2);
    cout << endl;
    cout << "Adding piece1 and piece2 gives" << endl;
    convr(piece1);
    cout << " + ";
    convr(piece2);
    cout << " = ";
    convr(total);
    cout << endl;
    return 0;
}

```

```

Length add(Length x, Length y)
{
    Length z;
    z.inches = x.inches + y.inches;
    z.feet = 0;
    if (x.inches >= 12.0)
    {
        z.inches -= 12.0;
        z.feet++;
    }
    z.feet += x.feet + y.feet;

    return z;
}

```

```

void print(Length k)
{
    cout << k.feet << " " << x.inches << " ";
}

```

- main() တွင် piece1 + piece2 = total ၏ structure type Length ၏ နာမည်ပေးခြင်း၊ create ပုံစံပါပဲ Length ၏ declaration ၏ program ထဲသို့ ပြန်လည်ထည့်သွင်းပြီး piece1 နှင့် piece2 ၏ member value တွင် အတိုင်းထည့်သွင်းပြီး add() function ကို call နေပါက argument တွင် piece1 နှင့် piece2 ၏ structure type ကို ပေးပါ။
- add() function တွင် ချက်ချင်းပင် piece1 ၏ x + piece2 ၏ y နှင့် assign ပြုလုပ်ပြီး x.inches ကို piece1.inches နှင့် ယူဆချက်ပေးပြီး z.inches = x.inches + y.inches = 7 + 11 = 18 ရှိပါက z.inches ၏ 0 နှင့် initialize ပြုလုပ်ပေးပြီး z.inches = 18 ကို 12 ထက်ကြီးသောကြောင့် အတိုင်းပေးပြီး 18 - 12 = 6 ဖြစ်ပြီး z.inches = 6 ဖြစ်ပြီး z.inches >= 12.0 မှာ 12 = 6 မှာ မှားသောကြောင့် z.feet ထည့်သွင်းပေးခြင်း မရှိဘဲ z.feet += နှင့် ထည့်သွင်းပေးပြီး z.feet = z.feet + 1 = 0 + 1 = 1 ရှိပါသည်။

- ဆရာတစ်ပါးက $z.feet = z.feet - x.feet + y.feet$ ဆိုတဲ့ expression ကနေ $1+25+56 = 81$ ကိုယူဆင်ယူပေးခြင်းပဲ။ အခုဆိုရင် structure z ကို $z.feet = 81$ နဲ့ $z.inches = 6$ ကိုပေးပြီးပေးရင်တော့ z ကို return ပေးပေးပေးနိုင်တဲ့ကုဒ်ကို $z = add()$ ပြင်ဆင်ပေးပြီး main() ကိုဆရာတစ်ပါးကပေးတဲ့ $total = add(piece1, piece2)$; ဆက်ပြင်ဆင်ပေးပြီး နံပါတ် total member ကုဒ်မှာ $total.feet = 81$ နဲ့ $total.inches = 6$ ဆိုတဲ့ နံပါတ်ကုဒ်ကြားပါ $convert(total)$ ကနေ အကြောင်းကုဒ်ပေးပေးနိုင်ရမယ်။ ဝ (၂) (၃) ကိုကြည့်ပါ။

```

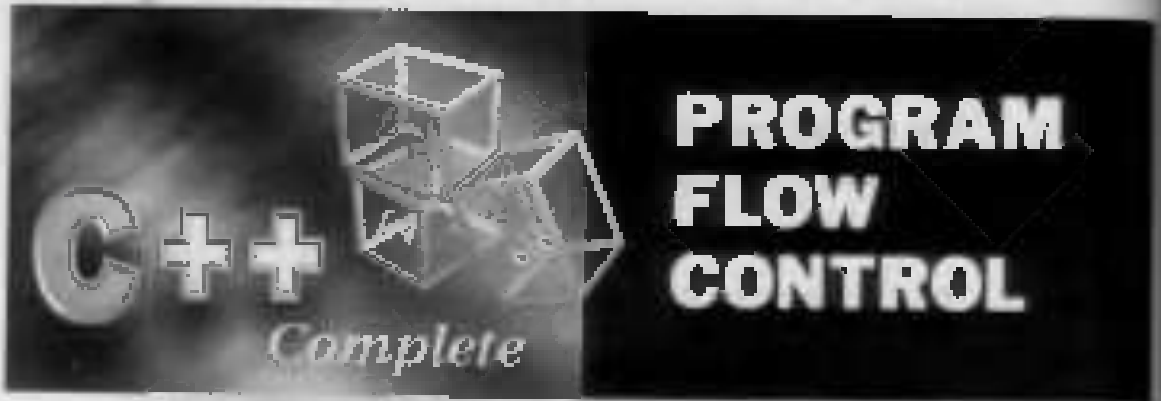
Quincy 99
For piece1
  Enter feet : 25
  Enter inches : 7

For piece2
  Enter feet : 55
  Enter inches : 11

Adding piece1 and piece2 gives
25'-7" + 55'-11" = 81'-6"

Any key to return to Quincy...
  
```

ဝ (၂) (၃)



C++ programming မှာ program မည်မှ, program flow ကို control လုပ်ပေးပုံစံတို့ကို
အသုံးပြု၍ C++ control statement ခေါ်သည်။ C++ မှာ အသုံးပြုသော statement များမှာ

break	do	if
case	else	return
continue	for	switch
default	goto	

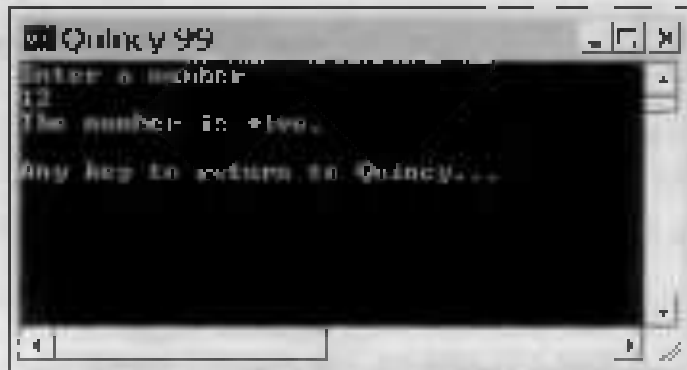
အောက်ဖော်ပြပါအတိုင်း အသုံးပြုသော စာလုံးများမှာ selection test များဖြစ်ပြီး if statement ဖြစ်သည်။

၃.၁ The if Statement

if statement သို့မဟုတ် conditional statement ခေါ်သည်။ if statement ကို အသုံးပြုသော မှတ်တမ်းများမှာ
if <test expr> မှတ်တမ်းများမှာ အောက်ဖော်ပြပါအတိုင်း ဖြစ်သည်။ အောက်ဖော်ပြပါ <test

ရဲ့အနိမ့်နားမှာ 32,768 ကနေ 32,767 အထိအမှားမရှိအောင်မယ်။ Enter a number ဆိုတဲ့ prompt ခေါ်လာတဲ့အခါမှာ num ကိုရက် data ကို 12 ရှိအောင်ထည့်ပြီး ENTER key နှိပ်ပေးရင် num = 12 ပြစ်သွားပါပြီ။

- ပထမဆုံး if statement ခွဲအောင်ပါသည်။ num က သုညထက်ပေါ်လာမယ့် ၁ ဘယ်ပယ်မဝပါး zero (FALSE) ကိုပယ်ပေးပေးနိုင်တဲ့အတွက် ကွန်ပျူတာက အောက်တစ်ကြောင်းကိုဆင်းသွားပေးပါမယ်။ if statement မှာ num က သုညထက်ကြီးလာမယ့် အခါကလည်း num = 12 က သုညထက်ကြီးလာမယ့်ကလေးပါ။ ဒီအတွက်ကွန်ပျူတာက <test expression> အောက်မှာရှိတဲ့ The number is +ive. ဆိုတဲ့တစ်ကြောင်းကို ကွန်ပျူတာမှာပင် လာလာထိကွပ်ပေးပေးပါသည်။ နှိုင်းကွပ်ရင် အောက်တစ်ကြောင်းထဲမှာဆက်တစ်ကြောင်းမှာပါ။
- အောက်ဆုံး If statement ကတော့ num ကို သုညနဲ့ညီလာတဲ့ပေးပါမယ်။ ပတ်ပါလား FALSE ပြစ်တဲ့ကွပ်ကွက် ကွန်ပျူတာက next line ကို သက်သက်သွားပါမယ်။ closing brace () နဲ့ ကွပ်ကွပ်အခါ program ခြိတ်သွားပါပြီ။ မို (ပုံ ၂) မှာ program ကို run ပြုလုပ်ပါသည်။



ပုံ (၁-၂)

Conditionally Excuting a Program Block

၁။ မိမိတို့ Ex301.cpp program ကိုပြင်ဆင်ပြီး အောက်တစ်ဆင့်အောင်ရတာက num က positive value ဖြစ်မှ num ကို သုညထက်ကြီးမယ်လို့ အခြေခံ display လုပ်ပြပါမိမ့်မယ်။ မဟုတ်ရင် You chose not to compare this message ကို display လုပ်ချင်မိလောက်ပါသည်။

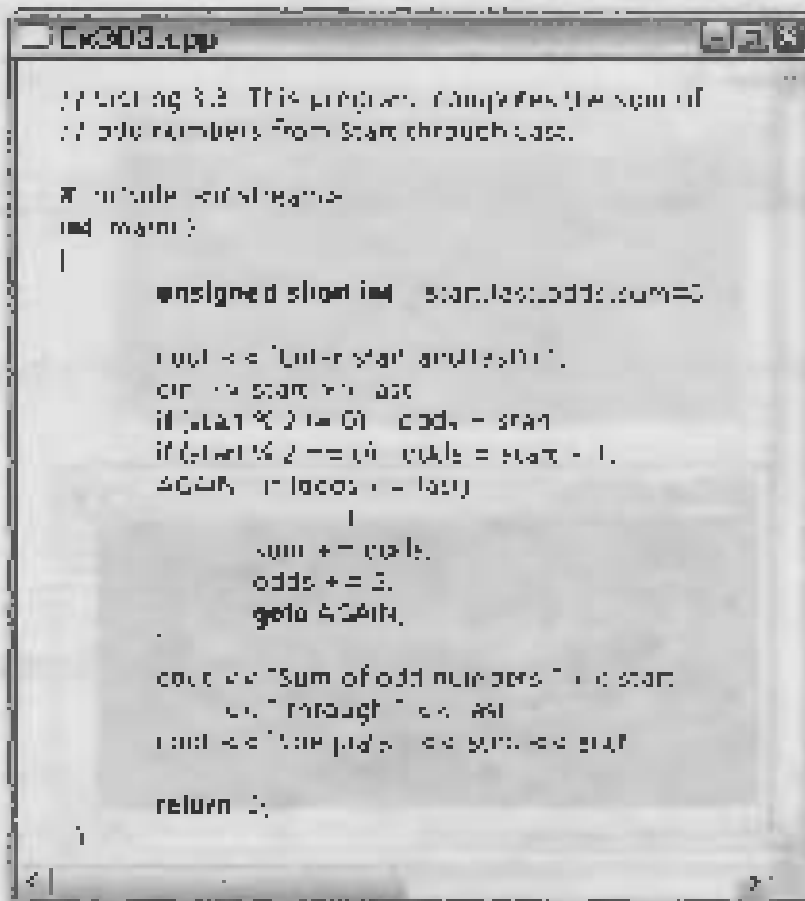
၃.၂ The goto Statement

goto သို့မဟုတ် program တစ်ခုခု ပြုလုပ်ပေးရန်အတွက်တစ်ခုတည်းသော အစဉ်အလာကို ရွေးချယ်နိုင်စေရန် အသုံးပြုသည့်အစဉ်အလာအမျိုးမျိုးကို အသုံးပြုရန် goto statement သို့မဟုတ် goto ချုပ်စားနိုင်ပါသည်။

```
<label> : target statement ;
```

```
goto <label> ;
```

label သို့မဟုတ် ရွေးချယ်ပေးရန်အစဉ်အလာအမျိုးမျိုးကို ရွေးချယ်နိုင်စေရန် အသုံးပြုသည့်အစဉ်အလာကို local label



```
Ex303.cpp
// Listing 3.3 This program computes the sum of
// odd numbers from start through last.

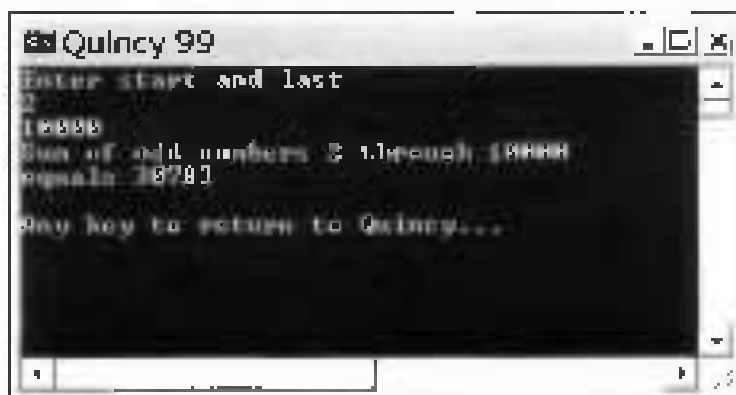
#include <iostream>
int main()
{
    unsigned short int start,last,odds,sum=0;

    cout << "Enter start and last:\n";
    cin >> start >> last;
    if (start % 2 != 0) odds = start;
    if (last % 2 == 0) odds = start + 1;
    AGAIN: if (odds <= last)
    {
        sum += odds;
        odds += 2;
        goto AGAIN;
    }

    cout << "Sum of odd numbers " << start
    << " through " << last
    << "\n\n";
    return 0;
}
```

၆ (၃.၄)

- goto statement ဟာ AGAIN နဲ့တဲ့ label name နဲ့တဲ့အခါကို ပြန် ပြောင်းနေတာပါ။ အခုဆိုရင် ကွန်ပျူတာဟာ block statement အတွင်းမှာ အိမ်ထဲထဲပစ်နေမှန်ဖြစ်ပါအောင် တစ်ချိန်မှာ odds = 11 ပြင်လွှာပါးပိန်မယ် မှီဆိုရင် odds = 11 ဟာ last = 9 ထက်ပိုကျွန်းတာကြောင့် <test expression> ဟာ TRUE ပြန်လွှာပိန်ပြီး if block ထဲကိုတစ်ခေါင်ပေလွှာပိန်တာကိုဆိုက်လာပါပြီခင်ဤ။
- အောက်ဆုံးမှာ Sum of odd numbers 2 through 9 equals 24 ဆိုတဲ့ output တွေကြောင့် ကို (2) ခံကြောင်းပြန် ပေးပြောပေးပါလိမ့်မယ်။ ပါပိုရ် Ex303.cpp program တွင်ပေးပို့ပြီး ပုံ (၃-၆) ကိုပြထားတဲ့ပုံတိုင်း start = 2 နဲ့ last = 10000 ကို data သွင်းရင် အောက်ဆုံးပုံပြင်ပေးပါ။



ပုံ (၃-၆)

Reversing an Integer Number

၁။ Ex304.cpp program မှဲ့စဉ်ရွယ်ချက်က ကိန်းပြည့်အကိန်းတစ်ခုကို input data အနေနဲ့ ထည့်ရင် ကိုက်ညီသမျှရင် အဲဒီကိန်းကို ပြောင်းပြန်အကိန်းပြန်ပေးတာ လုပ်ပေးတာပေးပါ။ ပုံ (၃-၇) မှာ test statement လေ့ကိုရေးပြထားပါတယ်။

၂။ Ex304.cpp program ကိုလေ့လာကြည့်မယ်ဆိုရင်

- အောက်မှာ unsigned long int num, rev, u, r (ရိုးရာတော့တဲ့)တို့မှာ * variable


```

num = q = 12345678
q = num / 10 = 12345678 / 10 = 1234567
r = num % 10 = 12345678 % 10 = 8
rev = rev*10 + r = 9*10 + 8 = 98

```

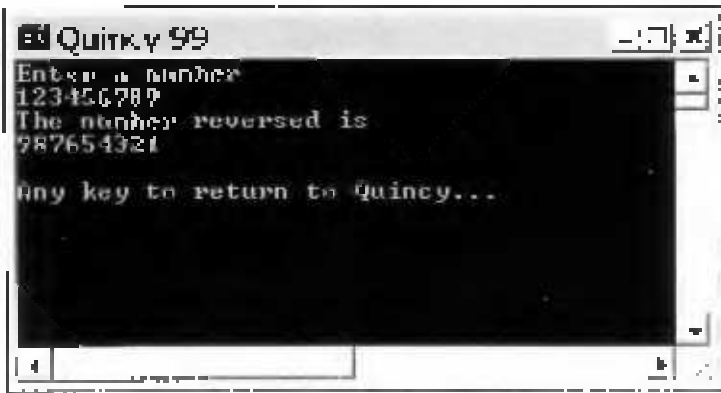
- goto START ဆိုက်ပ် ရွန်ပျူတာက START label ရဲ့နံပါတ်ကို ပြန်ရှာပါကယ် q က ဂျပေလက်စတီဆွတ် if block ထဲကို ရွန်ပျူတာက ပေးကမ်းစံနဲ့ပုံပြန်ပေးပါပြီ။ if block ထဲမှာ ဆုံးသို့မရောက်ပါ။

```

num = q = 1234567
q = num / 10 = 1234567 / 10 = 123456
r = num % 10 = 1234567 % 10 = 7
rev = rev*10 + r = 98*10 + 7 = 987
goto START :ရဲ့နံပါတ်

```

- တစ်ခိုန့်မှာ q = 0 ပြစ်သွားပြီး if block ထဲကထွက်လာမှာပါ။ ပေးကမ်းပေး The number reversed is : 987654321 သို့ တစ်ဆိုက်ပ်နဲ့စပြီး ကပြန်ကတ်ထုတ်ပေးပါ။မိမိမယ်။ ဝါဆိုရန် program ပြီးသွားပြီ။ ကပြန်ပုံ (a, b) မှာ program num ပြောင်းတယ်။



ပုံ (a- b)

A Payroll Program

Ex305.cpp program ရဲ့ ရည်ရွယ်ချက်က အလုပ်သမားတစ်ယောက်ရဲ့ လုပ်ငန်းပုံစံကိုယ်တိုင် ဆောင်ရွက်သော program ရှိပါတယ်။ အလုပ်ချိန် ၄၀ နဲ့ အလုပ်ချိန်ပိုမရဘဲ လုပ်ငန်းပုံစံကို ဆက်ဆံခွင့်ကွပ်ကဲပေးပေးဖို့ (၃, ၅) မှာ code statement ပေးချီးဖော်ပြထားပါမယ်။

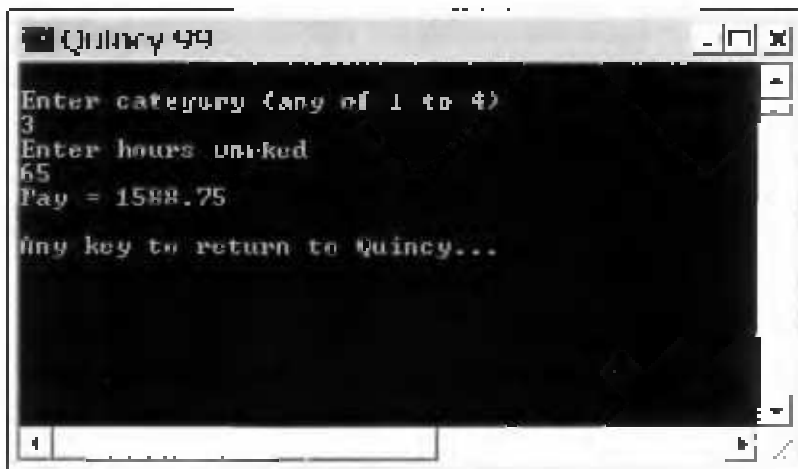
```
Ex305.cpp
// Listing 3.5. This program calculates the payroll for
// different workers giving overtime if the hours
// worked are greater than 40 hours.
#include <iostream>

int main()
{
    int    category;
    float  wage, hours, pay;

    START: cout << "\nEnter category (any of 1-4) or\n";
           cin >> category;
           if (category < 1 || category > 4)
               goto START;
           if (category == 1) wage = 12.5;
           if (category == 2) wage = 13.5;
           if (category == 3) wage = 20.5;
           if (category == 4) wage = 25.5;
           cout << "Enter hours worked:\n";
           cin >> hours;
           if (hours <= 40)
               pay = wage*hours;
           if (hours > 40)
               pay = wage*40 + 1.5*wage*(hours-40);
           cout << "Pay = " << pay << endl;
           return 0;
}
< |                                     > |
```

၃ (၃-၃)

၂။ ဤ program ကို ကွန်ပိုက်လ်ဆွဲယူပြီးနောက်မှ run ဆောင်ရာတွင် ပုံ (၃-၁၄) နှစ်ပါး ပြောင်းလဲသွားသော အဆင့်များကို ခြုံငုံပြီး data ပြောင်းလဲပုံကို run ကြည့်ပါ။ program ငါးခွက်ကြောင်း trace ဆွဲကြည့်ပါ။ မှတ်ပေးပါ။



ပုံ (၃-၁၄)

၃-၃ The if-else Statement

၁။ program ကို ရေးရာတွင် အချို့သော အခြေအနေအထား (2) ခုရှိရာတွင် if-else statement ကို သုံးစွဲရပါမည်။ If မှုအစစ်၊ <test expr> ကို အမှန်တရားရှိရင် <statement1> ကို အပြုဆောင်ပြီး if-else ကြိုတင်ဆိုတဲ့ အခြေအနေအထား <test expr> မှုအမှန်ရင် <statement2> ကို execute လုပ်မှာပါ။ statement မှုအား ကာရင်ကတ်ဆက်စိုလှောင် brace { } ခုတွင် စိတ်ဆောင်ပါမည်။ If-else statement: မျှဝေစား ဖွဲ့စည်းပုံ။

```

if ( <test expr> )
    <statement1>;
else
    <statement2>;

```

အဆင့်မြင့် . ပုံ (၃-၁၅) နှစ်ပါး ပြောင်းရာ Ex306.cpp program ကို if-else statement အသုံးပြု

Figure 3.5: A program that converts a lowercase letter to uppercase. The program first reads a lowercase character, then converts it to uppercase and then writes out the uppercase equivalent.

```

Ex306.cpp
// Listing 3.5 The program reads a lowercase character,
// converts it to uppercase and then writes out
// the uppercase equivalent.
#include <iostream>

int main()
{
    char lower, upper;

    cout << "Enter a lowercase character:\n";
    cin >> lower;
    if (lower >= 'a' && lower <= 'z')
        upper = 'A' + lower - 'a';
    else
        upper = lower;
    cout << "Uppercase equivalent is " << upper << endl;
    return 0;
}

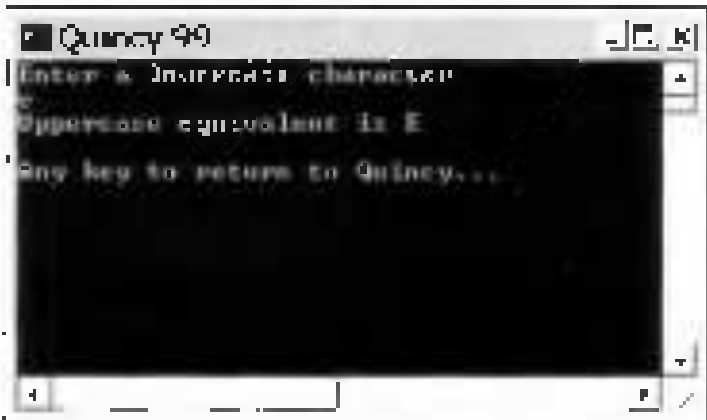
```

Figure 3.5

Ex306.cpp program အတိုင်းပြောဆိုပါသည်။

- program ကျောင့်မှ lower နှင့် upper ဆိုတဲ့ variable (2) မှာ character လေးခုပေးပြီး
 ပြောပါ။ Enter a lowercase character. ဆိုတဲ့ prompt ဆိုတဲ့ စာကိုပေးပြီး Enter ခဲ့ရင်
 lower မှာ data ကို စုစည်းပြီး ပေးပါ။ ပြီးရင် lower = 'a' ခြင်္သေ့ပေးပြီး
- if (lower >= 'a' && lower <= 'z') ဆိုတဲ့ relational expression ဆိုတဲ့ - lower ကို
 'a' ခြင်္သေ့ပြီး 'z' ကိုလည်းကောင်း စစ်ဆေးပါ။ TRUE ဖြစ်ရင် upper = A + lower -
 a ခြင်္သေ့ပေးပြီး upper ရဲ့ ASCII ကိုလည်း upper = 65 + 101 - 97 = 69 ခြင်္သေ့ပေးပြီး
 69 ကို ASCII ကိုလည်း 69 ကို uppercase letter E ဆိုတဲ့အတိုင်း ပြောတဲ့ စာကိုလည်း
 ပေးပါ။ ပြောပြီး ပြန်လည်စစ်ဆေးပါ။

- cout << "uppercase equivalent is " << upper; // statement #၉-ကို
 ကျန် E သုံးခုကလေးများ ခံ (၂၀-၂၅) ယူ program run [အားသာမှု] ပြုလုပ်ပါ မှတ်တမ်း
 keyboard ကနေ E ကို နှိပ်သည့်အခါတိုင်း ပုံနှိပ်ချက် အသံထိပ်ထုတ်ပေးပေး အဲဒါကို
 တွေ့ရသလိုပဲ နှိပ် လက်တွေ့အခြေအနေပြန်ပါ။



ပုံ (၉-၁၅)

A Temperature Conversion Program

၁၀-၂၀-၂၀၂၀-ခုနှစ်မှ program မှာ degree Fahrenheit ကနေ Celsius ကိုပြောင်း အချိန်တစ်ခုကို degree Celsius ကနေ Fahrenheit ကိုပြောင်း(ပြောင်းပစ်နိုင်) program မှာမှာပါ flag သတ္တိကို data ကို ၂ ခုကလေး မှားကလေးကို degree Celsius ကနေ Fahrenheit ကိုပြောင်းပြီး အားသာမှုကလေး ၇ နှစ်ကလေးကလေးကလေး degree Fahrenheit ကနေ Celsius ကိုပြောင်းပြီး အားသာမှုကလေး ၃၀၀၀၀၀ source code ကွန်ပို (၂) (၃) ယူ အခြေအနေကလေး အခြေအနေပြန်ပါ။

၂၀-၂၀-၂၀၂၀-ခုနှစ်မှ program run [အားသာမှု] ပြုလုပ်ပါ မှတ်တမ်း data ကို ၂ နှစ်ကလေးကလေး degree Celsius = 100 ကိုနှိပ်ကလေး ချက် Fahrenheit = 212 ကို display ထုတ်ပေး ပုံကို

```

Ex307.cpp
// Listing 3.7 A program that gives the user the option of
// converting Fahrenheit to Celsius or Celsius to Fahrenheit.
#include <iostream>

int main()
{
    int    flag;
    double tempr;

    cout << "Type 1 to convert Fahrenheit to Celsius. " << endl;
    cout << "Type 2 to convert Celsius to Fahrenheit. " << endl;
    cin >> flag;

    if (flag == 1) {
        cout << "Enter temperature in deg Fahrenheit. " << endl;
        cin >> tempr;
        cout << "Celsius = " << (tempr-32)/1.8 << endl;
    }
    else {
        cout << "Enter temperature in deg Celsius. " << endl;
        cin >> tempr;
        cout << "Fahrenheit = " << 1.8*tempr+32 << endl;
    }
    return 0;
}

```

☞ (p. 37)

```

Quincy 99
Type 1 to convert Fahrenheit to Celsius +
Type 2 to convert Celsius to Fahrenheit +
Enter temperature in deg Celsius + 100
Fahrenheit = 212
Any key to return to Quincy...

```

☞ (p. 39)

၃.၄ Nested if Statement

၁။ nested if-else statement ဟု if statement အောက်တွင်အခြေခံပါကမ်း စေ့ပြောင်းမှုကွဲလွဲမှု တွင်အတိတ်ပေးသောခွဲကွဲ if-else statement အတွက် ရှေ့အကမ်း (၂) ခုပေးထားပါကကျွန်ုပ်တို့ if <test expr> ဟု TRUE ဆိုရာတွင်အောက်က <statement> ကိုစေ့ပြောင်းပေး၊ FALSE ဆိုရာတွင် else မှုကမ်း အောက်တွင်ကျွန်ုပ်တို့ အခြေခံပြောင်းမှုအကမ်း (၂) မှုပေးထားပါကကျွန်ုပ်တို့ တစ်သက်သို့ သွားရောက်မည့် အမှန်တိုင်း ကို create လုပ်ဆောင်ရာတွင် nested if-else statement အတွက်အမှန်ပြုပေးပါသည်။ မှုမမှန်ဆိုရာတွင် brace ပေးပါ ကျွန်ုပ်တို့ ကျွန်ုပ်တို့ ပုံ (၃.၁၅) မှုပေးပါကြောင်းကို Ex308.cpp program ဟုဆိုရာတွင် complex nested if statement ဆို အမှန်ပြုပေးခြင်းကို မှုမမှန်အခြေခံအခြေခံပါသည်။

၂။ Ex308.cpp program ကိုအသုံးပြုဆောင်ရွက်မည်ဆိုရာတွင်

- program စေ့ပြောင်းရာတွင် num1 နှင့် num2 မှုပေးပါ integer variable အတွက်အခြေခံပါသည်။ Enter two numbers ဆို prompt ပေးပါသောအခါတွင် num1 ကို 13 ကိုညွှန်ပေးသောအခါတွင်၊ ခါဆိုရာတွင် num1 = 13 ဖြစ်သွားပါပြီး၊ num2 ကို 4 ကိုညွှန်ပေးသောအခါတွင် 4 ကိုညွှန်ပေးပါသည်။
- program က အခြေခံအောက်တွင် if statement မှု num1 = 13 ဟု num2 = 4 ကို ညွှန်ပေးပါသည်။ ဤသို့အခြေခံပါသည်။ <test expr> ဟု TRUE ဖြစ်ရာတွင် အောက်က if statement ကိုအောက်တွင် execute လုပ်ပါသည်။ အသုံးပြုပါကကျွန်ုပ်တို့ if statement မှုပေးပါက num1 = 13 ဟု num2 = 4 မှု ညွှန်ပေးပါသည်။ ၅ <test expr> ဟု FALSE ဖြစ်ရာတွင်၊ ဒီအခါ အောက်ကကျွန်ုပ်တို့ if-else ကိုအောက်တွင် ရက်က if-else မှုပေးပါက <statement> ကို execute လုပ်ပါပြီး They are not evenly divisible. ဆိုရာတွင် ကျွန်ုပ်တို့ display လုပ်ပါသည်။
- အမှန်ဆိုရာတွင် အခြေခံအောက်တွင် if-else မှုအောက်တွင်ပေးပါပြီး ဤရာတွင် closing brace ()) ဆိုရာတွင် မှုပေးပါ။ အောက်က program ကိုအသုံးပြုပါ။

၃။ ပုံ (၃.၁၆) မှု Ex308.cpp program ကို run ပြုထားပါသည်။ num1 = 13၊ num2 = 4 ဖြစ်ရာတွင် They are not evenly divisible. ဆိုရာတွင် message ကို display လုပ်ပါသည်။ num1 = 16၊ num2 = 4 ဖြစ်ရာတွင် သက်သို့ message ကို display လုပ်ပါသည်။ အောက်ကကျွန်ုပ်တို့ မှုပေးပါက program run ကိုအသုံးပြုပါ အောက်က ဖြစ်ပါသည်။

```
Ex308.cpp
// Listing 3.8 This program demonstrates a complex,
// nested if statement
#include <iostream>

int main()
{
    int num1, num2;

    cout << "Enter two numbers\nfirst : ";
    cin >> num1;
    cout << "Second : ";
    cin >> num2;
    if (num1 >= num2)
    {
        if (num1 % num2 == 0)
        {
            if (num1 == num2)
                cout << "They are the same.\n";
            else
                cout << "They are evenly divisible.\n";
        }
        else
            cout << "They are not evenly divisible.\n";
    }
    else
        cout << "The second one is larger.\n";
    return 0;
}
```

Figure 3-10

```
BBQuincy 99
Enter two numbers:
First : 13
Second : 4
They are not evenly divisible.
Any key to return to Quincy...
```

Figure 3-11

၃၂. Ex308A.cpp program မှ brace မွှေးထိုးခြင်းဖြင့် (၃-၁၇) မှာပြထားတဲ့ကနဦး ပြင်ဆင်မှုကို သုံးသပ်ပြီး run လုပ်ပါက အခြေသတ်အတိုင်းပေးပါ။ အောက်ဖွဲ့ဆိုက်ကတ်တွေက run ပြုနိုင်ပါ။

```

Ex308A.cpp
#include <iostream>

int main()
{
    int num1, num2;

    cout << "Enter two numbers: first ";
    num1 >> num1;
    cout << "Second ";
    num2 >> num2;

    if (num1 >= num2)
        if (num1 % num2 == 0)
            if (num1 == num2)
                cout << "They are the same.\n";
            else
                cout << "They are evenly divisible.\n";
        else
            cout << "They are not evenly divisible.\n";
    else
        cout << "The second one is larger.\n";

    return 0;
}

```

၃ (၃-၁၇)

၃၃. ဝါးမတုတ် program logic ကိုအတိုပြုသဖြင့် run လုပ်သော် အခြေအတိုင်းပေးပါ။ ဒီ program က compact ခြုံငုံရေးဆွဲကတ်ပေးပါ။ (၃-၁၈) ကိုကြည့်ပါ။

၃၄. Ex308B.cpp program ကို run ကြည့်ရင် (၃-၁၉) မှာပြထားတဲ့ပုံနဲ့ပြင်ရမယ်။ First = 15 & Second = 100 ကို data ထည့်သွင်းစမ်း။ The second one is larger. ကိုထပ်မံပြန် display လုပ်ပြပါ။ခိုင်မယ်။


```
Ex3088.cpp

#include <iostream>

int main()
{
    int num1, num2;

    cout << "Enter two numbers (first) : ";
    cin >> num1;
    cout << "Second : ";
    cin >> num2;

    if (num1 < num2)
        cout << "The second one is larger.\n";
    else if (num1 % num2 != 0)
        cout << "They are not evenly divisible.\n";
    else if (num1 != num2)
        cout << "They are evenly divisible.\n";
    else
        cout << "They are the same.\n";

    return 0;
}
```

Ⓢ (p. 88)

```
Quincy 99
Enter two numbers
First : 15
Second : 100
The second one is larger.
Any key to return to Quincy...
```

Ⓢ (p. 88)

Leap Year Test

• အခုတင်ပြဖူးတဲ့ Ex309.cpp program မှာဆိုရင် အင်္ဂလိပ် ဘယ်ခုနှစ်ဟာ leap year ဟုတ်သလား ဆိုတာကို စိစစ်ပေးရပါမယ်။ ခန့်မှန်းရာ ခုနှစ်ကို 4 ခုနစ်ရင်ပေးပြောပါ။ မကြိုက်ဘူးဆိုရင် အင်္ဂလိပ်ကတိုက်ကလေးပဲ ဘူး။ 4 ခုနစ်ပေးပြောရင် ငါ့ဘက်ဆိုင်ကလေးပါ။ ကလေးကလေးပေးပေး 200 ခုနစ်ပေးပြောပါ။ အင်္ဂလိပ်ပြောရင် အင်္ဂလိပ်ဟာ ပေးချာပေးကံ ခါးကလေးပါ။ ကြိုက်ကလေးဆိုရင် အင်္ဂလိပ်ကလေးပေးပေး။ မပေးပေးပေးပေးဘူးဆိုရင်ပေးချာပေး 400 ခုနစ်ပေးပြောပါ။ ပြောကလေးဆိုရင်ကလေးပဲ ကလေး။ ပြောရင်ကလေးပေးပေးပေးပေး။ အင်္ဂလိပ်ဆို program ကိုပေးပေးပေး ပြောပါ။ ခု (3, 36) က code statement ပေးပေးပေးပြောပါ။

```
ex309.cpp
// Listing 3.9 This program reads in a number representing
// a year and then determines whether it is a leap year or not.

#include <iostream>
const int TRUE=1,
const int FALSE=0,

int main()
{
    int year, leap;

    cout << "Enter a year (****)\n";
    cin >> year;

    if (year % 4 == 0) leap = FALSE
    else if (year % 100 == 0) leap = TRUE
    else if (year % 400 == 0) leap = TRUE
    else leap = FALSE;

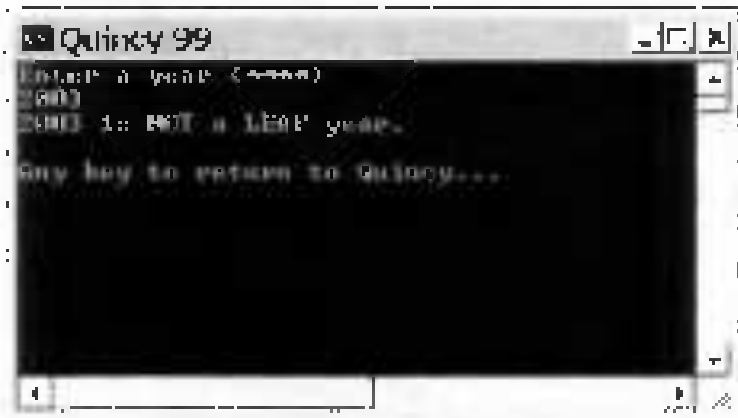
    if (leap) cout << year << " is a LEAP year.\n";
    else cout << year << " is NOT a LEAP year.\n";

    return 0;
}
```

(3, 36)

Ex309.cpp program တီထွင်ရေးအကြံပြုချက်များ

- 1998 ခုနှစ်ဟာ ဝါးယပ်ပါသည်။ ဘာမျိုးလွယ်ပဲဆိုတာ 1998 ကို 4 နှစ်စားစဉ်ပျက်လွယ်ပဲ 2000 ခုနှစ်ဟာ င်ယပ်ပါတယ်။ ဘာဖြင့်လွယ်ပဲဆိုတာ 2000 ကို 4 နှစ်စားစဉ်ပျက်တယ်။ 100 နှစ်စားစဉ်ပျက်တယ်။ ဒီဟော့ 400 နှစ်စားစဉ်ပျက်တယ် ဖြစ်ပေးတယ်။ ပါးမို့ပဲ 2000 ခုနှစ်ဟာ စားဖျာပါယပ်ပေးပါ။
- 5000 ခုနှစ်တွေဟာ ဝါးယပ်ပေးတာ။ ဘာဖြင့်လွယ်ပဲဆိုတာ ၁၀၀၀ 4 နှစ်စားစဉ်ပျက်တော့ 100 နှစ်စားစဉ်ပျက်တယ်။ ဒီဟော့ 400 နှစ်စားစဉ်ပျက်တော့ ၁၀၀ ပါးကြောင့်ပဲ 5000 ခုနှစ်ဟာ ဝါးယပ်ပေးလို့မဖြစ်တယ်။ ဒီနေရာမှာ nested if-else statement ကို အသုံးပြုထားပါကလည်း Ex309.cpp program ကို run ကြည့်ရင် ပို (၃-၂၀) မှာပြောထားတဲ့အတိုင်းပျက်ပေါ့။ year= 2003 လို့ထည့်ထားရင် 2003 is NOT a LEAP year. ဆိုတဲ့အဖြေကို display ထုတ်ပြန်လိမ့်မယ်။



ပုံ (၃-၂၀)

၃-၅ The while Statement

၁။ C++ program မှာ loop အစုံမှာကို ယာခါတစ်လဲပတ်ရတာပေးဖို့အတွက် while statement ကို အသုံးပြုလို့ရပါတယ်။ while statement မှဲ့ပဲစား အမှန်လို့ပါ။


```

Ex3010A.cpp
#include <iostream>

int main()
{
    int digit = 0;

    while (digit < 9; cout << digit << " " << endl)
        digit++;

    return 0;
}

```

ပုံ (၆-၂)

Evaluating PI Series

• ကနဦးတွင် Ex3011.cpp program တွေ့ရင် π ရဲ့ ဘဲနိန်းကို $3.14 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$ series ကို ပြောနေကြပြီး term ကို တွေ့ရဖို့ ခက်ခဲနေကြတာကို တွေ့ရတာပဲ။ ဒါမှတော့ program ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။ ဒါမှတော့ ဒီ program ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။ ဒါမှတော့ ဒီ program ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။

• Ex3011.cpp program ကို လိုက်ကြည့်ရင် အစမှာ pi = 0 ကို initialize လုပ်ပြီး n = 3000000 ကို ပေးထားပြီး term ကို တွေ့ရတာပဲ။ (၂) ဒါမှတော့ ဒီ series ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။ (၃) ဒါမှတော့ ဒီ series ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။ (၄) ဒါမှတော့ ဒီ series ကို ပြန်ကြည့်ရင် ဒီ series က slow converging ပြီးလို့ ခက်ခဲတဲ့ C++ compiler ကို ဖြစ်စေတာကို တွေ့ရတာပဲ။

```

Ex3011.cpp
// Listing 5.11 This program computes the value
// of pi using the series
//       $\pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$ 
//
// Compile and Run
#include <iostream>
using namespace std;
int main()
{
    double pi = 1, n = 1, sign = 1;

    cout << "Enter n: ";
    int n;
    while (cin >= 0)
    {
        pi += sign * 1.0/n;
        n += 2;
        sign = -sign;
    }
    cout << "pi = " << pi << endl;
    return 0;
}

```

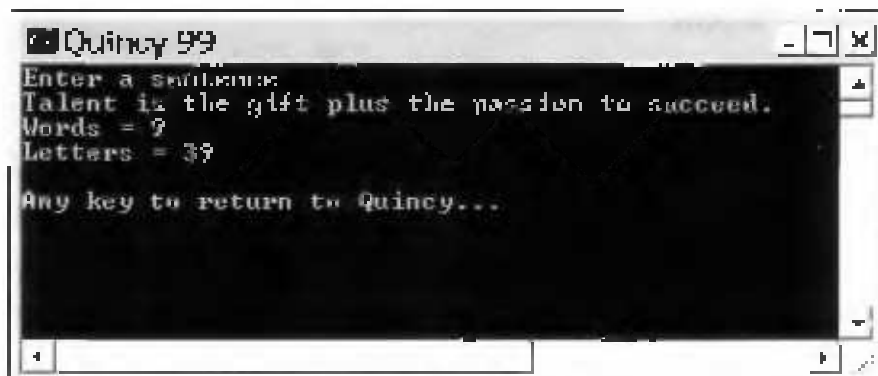
(a) (b)

```

Quincy 99
Enter n: 1000000
pi = 3.141592654
Any key to return to Quincy...

```

(c) (d)



ပုံ (၃-၂၈)

၃.၆ The do-while Statement

၁၁ C++ program မှ loop တစ်ခုကို while statement တို့ဖြင့်ပြုလုပ်ကြကော်လည်း ယခုကော်ပြီလောက်ပင် မတူဘဲ loop body ကို စစ်ဆေးပြီးမှသာ ကာလတစ်ကြိမ်ကို test နှုတ် loop ရဲ့အောက်တွင် ဆက်လက် ထည့်သွင်းနိုင်မည်။ အဲဒါကို do-while statement ဟုခေါ်ပြီး do-while loop ခုတ်လုပ်နိုင်ပါသည်။

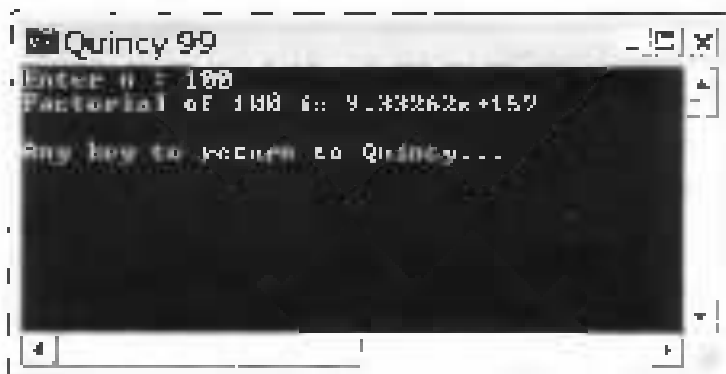
```

do {
    <statement1>;
    <statement2>;
    . . . . .
} while ( <test expression> );

```

ထိုသို့ loop ထဲသို့ အောက်ဖော်ပြပါအတိုင်း စတင်ပြီးမှ loop body ထဲသို့ အညွှန်းပါသည်။ loop body ထဲမှ statement တွေကို execute လုပ်ပြီးမှသာ while နှုတ်သည်။ ဒီလိုလုပ်ရင် while ကိုစတင်တဲ့ <test expr> ကိုစစ်ဆေးရင် TRUE ဆိုရင် do-while loop ကို လက်တင်အောက်တွင် ထည့်သွင်းပါ။ FALSE ဆိုရင် ထည့်သွင်းခြင်းမရှိပါ။ while ကာလတစ်ကြိမ်ကို ဖြည့်စွက်ပြီးမှသာ do-while control statement ရဲ့အောက်တွင် ဆက်လက် ထည့်သွင်းပါ။ do-while statement ကို လက်တွေ့က ဆုံးဖြတ်ကြည့်ရအောင်။

- ဆချင်း do-while loop တစ်ကြိမ်ပြီးတော့ $nfac = nfac * i = 1 * 2 = 2$ ကိုတွင်ယူတာက သွက်ငါ့ကား၊ နှိုင်းတာရင် 1 ကိုတစ်ကြိမ်ပါဝင်ပဲ။ do-while loop ကို $i > 5$ ပုံနှိုင်းပေးပြီးပယ်ပေးမှပါ။ $i = 6$ ဖြစ်လာတဲ့အခါကရင် do-while loop ကို ဆက်မပတ်တော့ပါဘူး။ next line ကို ဆင်သင်ပြီးတော့ Factorial of 5 is 120 ကိုကေးပြောတဲ့အခါကလဲနဲ့မယ်။ $n = 100$ ကာတွက်တုန်း run ကြည့်ပါဦး။ ဗုံ (၃-၃၀) ကိုကြည့်ပါ။ ဒါက do-while statement ကိုကေးပြောတာပါ။



ဗုံ (၃-၃၀)

Prime Factors of a Number

၁။ အခုအင်ဂျင်နာ Ex3014.cpp program ခဲ့ရင်ရှာဖွေချက်က အကဲခတ်လျက်ကော့ရင် အကဲခတ်ရင် ဆွဲဆွဲနဲ့ကေးတွေကို ဆေးဆေးတဲ့ program ဖြစ်ပါတယ်။ ဗုံ (၃-၃၀) ကိုကြည့်ပါ။

၂။ Ex3013.cpp program ကိုအခုအင်ဂျင်နာအင်အင်နာရင်

- program အများ Enter a number ကို prompt ပေါ်လာတဲ့အချိန်မှာ 105 ကိုနိုက်တာရင်ပေး ကိုကေးရင် $n = 105$ ဖြစ်ပွားမိပြီး divisor = 2 adder = 1 ကို initialize လုပ်ထားရင် အဲဒါက do-while loop ကိုပေးမိပြီး if statement ခဲ့ $n \% \text{divisor}$ ပေးအများအင်ဂျင်နာပေါ်လာတာ 105 ကို 2 နဲ့ပေးရင် အခြေခံလုပ်ပေးရင်ပေး၊ ဒီထက် else ပေးအင်ဂျင်နာပေါ်ရင် ပေးနိုယူတာ လုပ်မှပါ။ divisor ကို 1 ကိုကေးရင် adder = 2 ကို assign လုပ်ပါကယ်။ divisor = 3 ဖြစ်ပွားမိပြီး ဒါပေးရင် $n = 105$ က ပေးပြောပေးပေးရင်ပေး။

```

EK3014.cpp
// Listing 3.14 This program prints a list of
// the prime factors of a number n
#include <iostream>

int main()
{
    int divisor, adder = 1, n;

    cout << "Enter a number:\n";
    cin >> n;
    cout << "Prime factors of " << n << " are:\n";
    do {
        if (n % divisor == 0)
        {
            n /= divisor;
            cout << divisor << " * ";
        }
        else
        {
            divisor += adder;
            adder = 2;
        }
    } while (n / 2 > 1);
    cout << "\n";

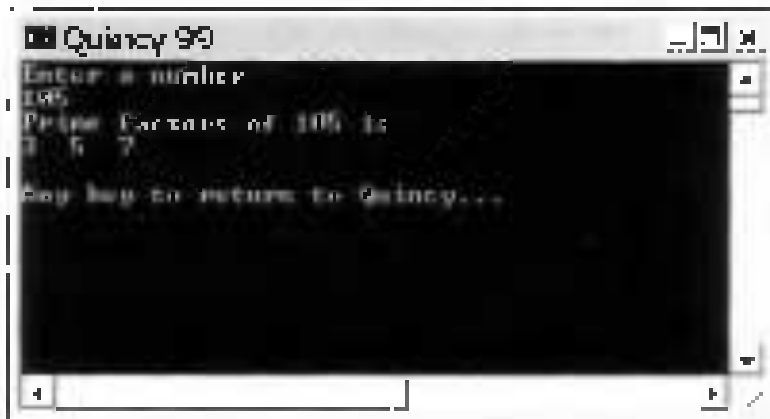
    return 0;
}

```

Figure 3.20

- while ($n/2 > 1$) ဆိုတာနဲ့ ဆိုက်, $n/2$ က 1 ကိုက်တာနဲ့ do-while loop ကို ရှိနေတဲ့ အား $n/2 = 105/2 = 52$ ဆိုတာ, 1 ကိုက်တာနဲ့ while loop ကို ရှိနေတဲ့ အား if ($n \% \text{divisor} == 0$) ဆိုတာ, 105 ကို divisor = 3 ဆိုတာနဲ့ ပေးနေတာ, $n = \text{remainder} = 105/3 = 35$ ဆိုတာနဲ့ n ကို 3 ဆိုတာနဲ့ ရှိနေတာ, $n = 35$ ဆိုတာနဲ့ display ပြုနေတာ
- $n/2 = 35/2 = 17$ က 1 ကိုက်တာနဲ့ do-while loop ကို ရှိနေတဲ့ အား 35 ကို 3 ဆိုတာနဲ့ ပေးနေတာ, else $\text{adder}++$; divisor = divisor + adder = 3 + 2 = 5 ဆိုတာနဲ့

while ဆိုရာကတည်းက $n/2 = 35/2 = 17$ ကိုဖြည့်သည့် do-while loop ကို
 [၃] ဖြစ်သည့် 35 ကို 5 နှင့် 7 နှစ်ဖြတ်ရာ $n = n/divisor = 35/5 = 7$ ကိုဖြည့်ရာမှ
 ကို 5 နှင့် 7 ကို display ပြုနိုင်သည့် ကိစ္စကတည်းက n ကို 105 ရှိသောနေရာမှ
 3 5 7 (၃) ကို အခြေအနေအထားမှ (၃, ၅, ၇) မှာ run ပြုသော output ကိုကြည့်ပါ။



၆ (၃-၅)

Linear Regression Method

၁၀) အောက်ဖော်ပြပါ Ex3015.cpp program ကိုဆိုင်ရာ မူလအား x - y data array (5) နှစ်ကိုအသုံးပြု၍
 error သေးသည့် ရှိသည့် နေရာတွင် equation ကိုရှာသောအခါ ရှိသည့် linear regression သို့မဟုတ်
 ရှိသည့် data point ကိုရှာသော ရှိသည့် equation ကိုရှာသောအခါ ရှိသည့် equation
 ကို ရှာသောနေရာတွင် ရှိသည့် equation ကို $Y = a + bx$ ရှိသည့်နေရာတွင် a နှင့် b ကို

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$a = \frac{\sum y - b(\sum x)}{n}$$

ဆိုတဲ့ equation တွေကို ဘက်စုံကွဲကွဲလဲလဲနဲ့ ပြန်ပို့နိုင်အောင်ပါ။ ဝေဖန်တဲ့ data pair ကောင်းကောင်းကို တွေ့
 program မှာ $n = 5$ လိုအပ်အောင် ပြောင်းလဲထားပါ။ program ကို general ပြုစုပေးရန်မို့ n ကို အပြောင်းအလဲထားပါ။
 program ကို အသုံးပြုသူကွဲကွဲနဲ့ ပြောင်းလဲအောင် ပြုစုပေးရန်မို့ ဘက်စုံကွဲကွဲလဲလဲနဲ့ ပြောင်းလဲပြုစုပေးပါ။

```

Ex2015.cpp
// Using 3.15 This program reads in pairs of data
// values x and y, calculates and prints the values of
// the regression coefficients a and b.
#include <iostream>

int main()
{
    int    n = 5;
    double x, y, sumx, sumy, sumxx, sumxy, a, b;

    sumx = sumy = sumxx = sumxy = 0;
    cout << "Enter x and y data arrays:\n";
    do
    {
        cin >> x >> y;
        sumx += x;
        sumy += y;
        sumxx += x*x;
        sumxy += x*y;
        ++n;
    } while (n <= n);

    // This is the regression formula
    b = (n*sumxy - sumx*sumy) / (n*sumxx - sumx*sumx/n);
    a = (sumy - b*sumx) / n;

    // Print result
    if (a > 0) cout << "y = " << a << "x + b\n";
    else cout << "y = " << a << "x + b\n";
    return 0;
}

```

၅ (၁၀ နှစ်)



Figure 2-1

2.7 The for Statement

- for statement ဆိုတာ C++ မှာ looping လုပ်ဖို့အတွက် အသုံးပြုတဲ့ control statement တစ်ခုပါ။ 3 statement များကလေးပါ။

```

for ( [<initialization>] ; [<test expr>] ; [<increment>] )
{
    <statement1>;
    <statement2>;
    .....
}

```

မူရင်း <initialization> expression က for loop ရှင်းတဲ့ counter ရှိပါတယ်။ Inital index ကို သတ်မှတ်ပေးတာပဲ။ <test expr> က loop ရှည်နေမယ့်အထိ မလုပ်နိုင်တော့မှ ရပ်တည်းပေးတဲ့ ပုံစံ။ 3 conditional expression ကလေးနဲ့ looping လုပ်နေတဲ့အတွက်ပေးတာပဲ။ <increment> expression က counter ကို increment ရှိမယ့်အထိ decrement လုပ်ပေးတာပဲ။ increment operator


```

Ex3017.cpp
// Listing 3.7: This program converts the temperatures
// in Celsius to Fahrenheit.
#include <iostream>

int main()
{
    signed int cel;
    float fah;

    cout << " CELSIUS    FAHRENHEIT\n";
    for (cel = -40; cel <= 40; cel += 10)
    {
        fah = 1.8*cel+32.0;
        cout.setf(ios::fixed), // set cout flag for fixed-point
        cout.width(10), // print width specified
        cout.precision(2) // specify two decimal places
        cout << cel << "C\n";

        // width() manipulator does not stick from one
        // command to the next
        cout.width(10);
        cout << fah << "F\n";
    }
    return 0;
}

```

ပုံ (၃-၃၉)

- for loop မှ `<initialization>` expression မှ `cel = -40` လိုက်လာပါက `fah` ဝင်အဖြစ် လိုက်မည်ဖြစ်ပြီး `fah = 1.8*cel+ 32 = 1.8*(-40)+ 32 = -40` လိုက်ကပ်လိုက်ပါသည်။ for loop အတွင်းမှာ `cel` ကို print ထပ်ပါက `tab` ဖြစ်ပါသည်။ `fah` ကို print ထပ်ပါက ကာလကို print လုပ်ပြီးနောက် closing brace () နှင့်တွဲ၍ `cel` ကို 10 ထိုးပါသည်။ `cel = -40+10 = -30` ဖြစ်သွားပါပြီး `cel = -30` မှ last counter ခြံမီ `cel = 40` နှင့်အညီပင် ထပ်လက်ပြီးနောက်ပိုမိုမည်တည်း။ `if <test expr>` မှာ TRUE ဖြစ်ကား for loop အားပင် ထပ်ပါကား။ တစ်ချိန်မှာ `cel=50` ဖြစ်လာမှာပါ။ `if` မှာ `cel=50` မှ 40 ထက်ပိုသော သော့မည်။ `if` ကျေကပ် FALSE ဖြစ်သွားပြီး for loop ထပ်မံမစတင်ကျေပါဘူး။ program output ကို ပုံ (၃-၃၉) မှာပါပြထားပါကား။



ပုံ (၃-၃၈)

၂ program ကဲ့သို့ setw() manipulator အသုံးပြုမှု အားဖြင့် ပြင်ဆင်ပြီးနောက် run မှတ်ချက်များ တွဲဖက်

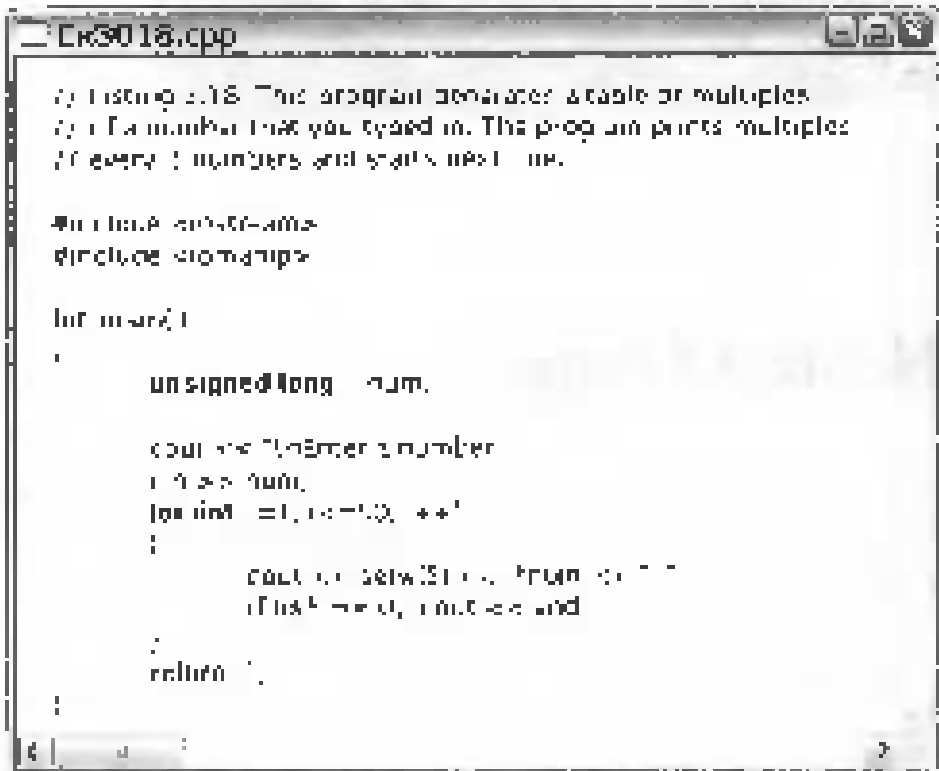
```
#include <iostream>
#include <iomanip>
```

```
int main( )
{
    signed int cel;
    float fah;

    cout << " CELSIUS FAHRENHEIT\n",
    for (cel = -40; cel <= 40; cel += 5)
    {
        fah = 1.8*cel + 32.0;
        cout << setw(10) << cel << setw(14) << fah << endl;
    }
    return 0;
}
```

Generating a Table of Multiples of a Number

၁၉. ကျွန်ုပ်တို့၏ Ex3018.cpp program ကာဘီဒင့် user ကို input ကိုရောင်း ဘက်ဆိုင်ရာ သုံးစွဲနိုင်ရေးကို (5) ကိုကော်မန်တီများကို display ကိုညွှန်ပေး program ဝန်ပြုပေးပါ။ for loop သုံးစွဲပြီးနောက်ပေးသည့်အတိုင်း (၃, ၃၃) ကို program ကိုကော်မန်တီပါ။



```
Ex3018.cpp
// Listing 3.18 This program generates a table of multiples
// of a number that you typed in. The program prints multiples
// every 5 numbers and starts next line.

#include <iostream>
using namespace std;

int main()
{
    unsigned long num;

    cout << "Enter a number: ";
    if (num == 0) num = 1;
    for (int i = 1; i <= 10; ++i)
    {
        cout << setw(5) << i << " from " << num << " = "
            << i * num << endl;
        if (i % 5 == 0) cout << endl;
    }
    return 0;
}
```

Figure 3.18

၂၀. Ex3018.cpp program ကို run ပြန်ဆိုရာ (၃, ၃၃) ကို ဝန်ရောင်းဘက်ဆိုင်ရာများကို 12 ကိုရောင်း 50 ကိုကော်မန်တီများကို ဝန်ရောင်းပါ။ (5) ကိုကော်မန်တီများကို next line ကိုကော်မန်တီ (i % 5 == 0) cout << endl, statement ကိုကော်မန်တီပါ။



ပုံ (၃.၄၁)

၃.၀ Nested Loops

၁။ loop တစ်ခုထဲတစ်ခုမှာ loop တစ်ခု သို့မဟုတ် တစ်ခုထပ်တစ်ခု loop ထွက်တဲ့အားကို nested loop ကိုခေါ်ပါသည်။ inner looping မှ outer looping ကို control structure ကိုသုံးသပ်သည့်အား တွေကို မဆိုပါဘူး inner loop မှ for မှတ်တမ်း၊ outer loop မှ while သို့မဟုတ် do-while ကိုသုံးသပ်ပါသည်။ ထိုသို့မဟုတ် outer loop မှ do-while မှတ်တမ်း၊ inner loop မှ for ကို သုံးသပ်သည့်အားဖြစ်ပါသည်။ ထိုသို့မဟုတ်မဟုတ် မှတ်တမ်း loop တစ်ခုထပ်တစ်ခုမှာ crossing ဖြစ်နိုင်ပါသည်။ ဤကဲ့သို့မဟုတ် loop က ထွက်ပါ loop ကို ထွက်ခွင့်မရပါ။

၂။ ကုမ္ပဏီကော်ပိုရေးရှင်း Ex0019.cpp program က 2-equation (3-unknowns) ကို nested for loop သုံးသပ်ပြီး ရှေ့ခိုင်းထားတာပါ ပုံ (၃.၄၁) မှ source code ထွက်ခွင့်မရှိထားပါ။

၃။ Ex0019.cpp program ကိုမဟုတ်အခြေအနေအထား

- program ကို program ကိုသုံးသပ် variable ထွက်ခွင့်မရှိ integer ကိုသုံးသပ်ပါသည်။ outer for loop မှတ်တမ်း k = 1 မှ inner for loop မှတ်တမ်းလက်ထက် 5 ကတည်း မှ 1 ကိုသုံးသပ်ပါသည်။

```

Ex019.cpp
// Listing 2.13 This program determines the three
// unknowns x, y, and z, which are defined by two equations:
//      x + y + z = 25
//      2.5x + 5y + 0.25z = 25
// where x, y, and z are integers.

#include <iostream>

int main()
{
    int x, y, z;

    for (y=1; y <= 10; ++y)
        for (x=1; x <= 5; ++x)
        {
            z = 25 - x - y;
            float sum = 2.5*x + 5*y + 0.25*z;
            if (sum == 25) goto RESULT;
        }
    RESULT: cout << "X = " << x << " Y = " << y << endl;
    cout << "Z = " << z << endl;
    return 0;
}

```

ပုံ (၃.၁၀)

- $z = 25 - x - y = 25 - 1 - 1 = 23$
 $\text{float sum} = 2.5 * x + 5 * y + 0.25 * z$
 $= 2.5 * 1 + 5 * 1 + 0.25 * 23 = 11.25$
- $\text{sum} = 25$ မဟုတ်သေးပါက inner loop ရဲ့ closing brace မှတ်၊ $\text{sum} = 13.25$ မှာ 25 နဲ့မတူသေးပါဘူး။
- $x = 1$ နဲ့ $y = 2$ သို့မဟုတ် $x = 2$ နဲ့ $y = 1$ တို့မှာ $\text{sum} = 25$ ဖြစ်သော်လည်း RESULT label မှတ်မရောက်ဘူး။

The switch Statement

statement group corresponding variable group performs different statement group depending upon group value. switch. flow statement. `switch` syntax is following.

```

switch ( <switch variable> )
{
    case <constant expr1> :
        <statement1>;
        [break;]
    case <constant expr2> :
        <statement2>;
        [break;]
    . . . . .
    . . . . .
    . . . . .
    default :
        <statement_default>;
}

```

Flow: switch expr1. flow to <switch variable> & current value. <constant expr1> & flow to <statement1> & execute. & break. & using brace { } & & break is optional. & flow to <switch variable> & current value & <constant expr2> & flow to <statement2> & execute. & using brace & & <constant expr> & eg. & & default & & <statement_default> & execute. & & switch blank & & & Ex3021.cpp program & & &

```

Ex3021.cpp
// Listing 3.21: This program calculates the payroll for
// four different classes of workers.
#include <iostream>

int main()
{
    int    category;
    float  wage, hours, pay;

    START: cout << "Enter category (any of 1 to 4)\n";
    cin >> category;
    switch (category)
    {
        case 1:
            wage = 12.5, break;
        case 2:
            wage = 15.5, break;
        case 3:
            wage = 20.5, break;
        case 4:
            wage = 25.5, break;
        default: ! cout << "Try again!\n";
                goto START;
    }

    cout << "Enter hours worked\n";
    cin >> hours;
    if (hours <= 40) pay = wage*hours;
    else pay = wage*40 + 1.5*wage*(hours-40);
    cout << "Pay = " << pay << endl;
    return 0;
}

```

ပုံ (၃၀၅)

Ex3021.cpp ကို run လုပ်တယ်ဆိုရင် ပုံ (၃၀၅) အခြေအနေအထားဖြစ်မှန်း category သေတ္တာ 3 & working hours သေတ္တာ 65 တို့ကိုရင်တည့်ပေးမယ်ဆိုရင် Pay = 1588.75 တို့တွေကပေါ့ဦး။ သေတ္တာ category < 1 တို့သေတ္တာ category > 4 ဖြစ်မယ် data ဖြစ်သတိပေးမိဦးမယ်။

```

Quincy 99
Enter a value from 1 thru 4: 3
Enter hours worked: 25
Pay = 1500.00
Any key to return to Quincy...

```

Figure 9-10

Use a switch Statement to Control a Menu

In the next program, `Ex022.cpp`, we use a `switch` statement to control a menu. We create a simple menu with three choices. The program will run until the user chooses choice 4, which will cause the program to exit.

```

Quincy 99
19. EXPERT'S COMPUTER REPAIR CENTER
What work performed? Here's the choices:
1. Replaced the CPU and RAM
2. Replaced the RAM only
3. Repaired the monitor
4. Fixed stuck keys
What work was performed?
Total charge is $350.00
Any key to return to Quincy...

```

Figure 9-11

```

hw3022.cpp
// Listing 8.22 This program uses a switch statement to
// control a menu's selections.
// Ming, use vi or emacs

int main()
{
    int    choice;
    float  charge=0;

    cout << "PC Lab PC's Computer Repair Center\n";
    cout << "Service performed: Here's the choices:\n";
    cout << "1) Replaced the CPU and RAM\n";
    cout << "2) Replaced the RAM only\n";
    cout << "3) Repaired the monitor\n";
    cout << "4) Fixed the keyboard\n";
    do
        cout << "What was performed? ";
        choice=0;
    while (choice < 1 || choice > 4)

    switch (choice)
    {
        case 1: charge = 200; // Notice no break here
        case 2: charge += 150; break;
        case 3: charge = 75; break;
        case 4: charge = 12;

        //
    }

    cout << "is\n";
    cout << "show price.\n";
    cout << "fixed.\n";
    cout << "Total charge is $ " << charge << "!\n";
    return 0;
}

```

(Figure 8.6)

In the program, the case 1 statement does not have a break statement. As a result, the switch (choice) statement will execute the case 1 statement, then the case 2 statement, then the case 3 statement, and finally the case 4 statement. This is because the switch loop will continue to execute until it reaches a break statement.

နောက်ဆုံး အသစ်တစ်ခု change + = 150 ရှိမှတ် change = 200+150 = 350 ရှိခဲ့ကြောင်းတွေ့ရပါသည်။
 break ကိုလည်းကောင်း switch loop ဆန့်ကျင်ဘက်ဘက်မှာ choice=2 ကို data input ပေးခဲ့သော တစ်ခု
 change = 150 ကိုလည်းကောင်း

၃.၃၀ The break Statement

break သုံး၍ looping သို့မဟုတ် switch ကိုရပ်စဲပေးပုံ exit ကိုပုံစံသွင်းပုံကဲ့သို့ မှတ်စု + while + do-
 while + for + switch statement ကဲ့သို့ break ကိုပုံစံသွင်းပုံကဲ့သို့ပါဝင်ပါသည်။ I statement ကဲ့သို့ exit
 ကိုပေးနိုင်သော break ကိုပုံစံသွင်းပုံကို ၀၀၁၀ နံပါတ်နံပါတ် Ex3023.cpp နှင့် Ex3020.cpp program

```

Ex3023.cpp
// Listing 3.23 This program decerns the three
// unknowns two. Actual Ans.
#include <iostream>

int main()
{
    int    x,y,z;
    float  sum;

    for (x=1; x <= 10; ++x)
    {
        for (y=1; y <= 5; ++y)
        {
            z = 25 - x * y;
            sum = 2.5*x + 5*y - 0.25*z;
            if (sum == 25) break;
        }
        if (sum == 25) break;
    }
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "z = " << z << endl;
    return 0;
}
    
```

၃.၂၉

ဆို break statement သည် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးပါသည်။ break statement ကို အကွက်ထွက်ခွင့်ပေးရန်အတွက် အသုံးပြုပါသည်။ break ကို အသုံးပြုရာတွင် ကျွန်ုပ်တို့သည် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးရန်အတွက် break ကို အသုံးပြုပါသည်။ break ကို အသုံးပြုရာတွင် ကျွန်ုပ်တို့သည် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးရန်အတွက် break ကို အသုံးပြုပါသည်။

The Equivalent of a Four-function Calculator

ကျွန်ုပ်တို့သည် break statement ကို အသုံးပြုရာတွင် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးရန်အတွက် break ကို အသုံးပြုပါသည်။ break ကို အသုံးပြုရာတွင် ကျွန်ုပ်တို့သည် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးရန်အတွက် break ကို အသုံးပြုပါသည်။ break ကို အသုံးပြုရာတွင် ကျွန်ုပ်တို့သည် ခြေထောက်မှ အကွက်ထွက်ခွင့်ပေးရန်အတွက် break ကို အသုံးပြုပါသည်။

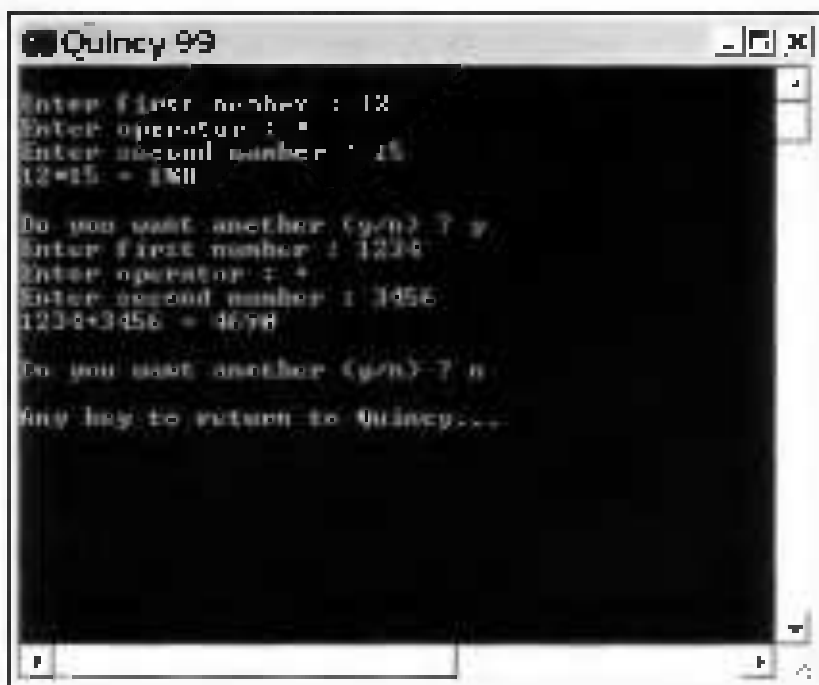


Figure 9.9

```

E:G024.cpp
// Listing 9.24 This program creates the equivalent of
// a built-in function calculator. The program will request
// the user to enter a first number, an operator, and another number.

#include <iostream>
#include <conio.h>

int main()
{
    double  num1, num2, ans;
    char    oprch;

    do {
        cout << "Enter first number: "; cin >> num1;
        cout << "Enter operator: ";   cin >> opr;
        cout << "Enter second number: "; cin >> num2;
        switch (opr) {
            case '+': ans = num1 + num2; break;
            case '-': ans = num1 - num2; break;
            case '*': ans = num1 * num2; break;
            case '/': ans = num1 / num2; break;
            default: ans = 0;
        }
        cout << num1 << opr << num2 << " = " << ans << endl;
        cout << "Do you want another (y/n)? ";
        ch = getch();
    } while (ch != 'n');

    cout << endl;
    return 0;
}

```

Figure 9.24

9.33 The continue Statement

Loop ကို မကြိုက်သော အချက်များကို loop ထဲက statement မှာရှောင်ကျွန်းပြီး execute လုပ်ပေးရန် အသုံးပြုနိုင်

loop မှာ ကိန်းတန်းတိုင်းကို loop ထဲ မှုတ်တင်ပြီး [for] [while] [do-while] [continue] ဆိုတဲ့ statement တို့ကို
 ရှိသလို၊ continue ကို while + do-while + for ကို သုံးသုံးသုံးသုံးလောက်

```

Ex3025.cpp
// Using 1.25 This program finds the average of the
// nonnegative integers from a list of numbers
#include <iostream>

int main()
{
    int    n, navg=0;
    float  x, avg, sum=0;

    cout << "How many numbers? ";
    cin >> n;
    for (int count=1; count<=n; ++count)
    {
        cout << "X = ";
        cin >> x;
        if (x < 0) continue;
        sum += x;
        ++navg;
    }
    avg = sum / navg;
    cout << "The average is " << avg << endl;
    return 0;
}
    
```

(Ex-25)

Ex3025.cpp program ကို သုံးသုံးကြည့်ရင် ဆိုရင်

- 3 program မှာ ကွဲလွဲတတ်တဲ့အချက်ကို ကို data argument, five number ကို နှစ်ထပ်
 ခေါ်ဆိုတာ၊ average ကို သုံးသုံးပေါ့ပေါ့၊ အသုံးပြု variable ကို type declare
 သတ်မှတ်၊ အဲဒါကို initialize လုပ်တာ၊ အသုံးပြုတာ၊ data မှုတ်တင်၊ prompt လိုက်
 data ကို သုံးတာ။

ထိုမှတစ်ဆင့် variable +> function ဆိုရာတွင် သို့မဟုတ် သတ်မှတ်ထားသည့် address ++ရာထုထွက်ပေးပါ။

၄.၁ Referencing Variables with Pointers

- (၁) (၄.၁) မှာပါ Ex401.cpp program ကို အောက်ဖော်ပြပါ char, int, float type variable
- (၂) ဆိုရာတွင် data value +> pointer ဆိုရာတွင် printf သုံးပြီး +> နှင့် ပုံစံပြောင်းလဲပေးပါ။

```
EX401.cpp
// Listing 4.1 Referencing variables with pointers
#include <iostream>

int main()
{
    // int size variable
    char c = 'A',
    int i = 123,
    float f = 567.89;

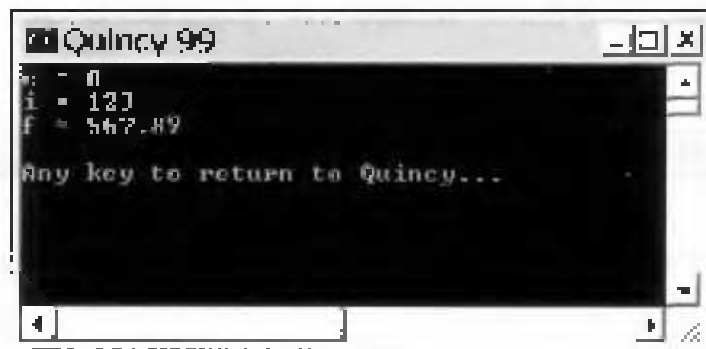
    // define pointer types and int size with address
    char* cp = &c;
    int* ip = &i;
    float* fp = &f;

    // print variable values using pointers
    cout << "c = " << *cp << endl;
    cout << "i = " << *ip << endl;
    cout << "f = " << *fp << endl;
    return 0;
}
```

— (၄.၁)

Ex401.cpp program ကိုယ်စားပြုသည့်မည်သို့ရေး

- ပထမဦးစွာ c, i နှင့် f ဆိုတဲ့ variable (3) မှာတွက် data type ဆွဲ၊ ခွဲခြားခြင်းကို သတိပေးကာ အသုံးပြုနိုင်အောင် initialize လုပ်ပေးပါသည်။ ဆက်လို့ pointer (3) ချိတ်ဆွဲတဲ့ cp နှင့် fp ဆိုတဲ့ define လုပ်ပြီး c, i နှင့် f variable ဆွဲရုံ address value ဆွဲရုံ initialize လုပ်ပေးပေးတဲ့ variable c ထဲက 'A' ကိုလှည့်ရင် "cp ကို display လုပ်နိုင်ရပါမည်။
- ဒီနည်းတူတိုင်း i နှင့် f ကိုလှည့် value ရှေ့ထိုးလှည့် display လုပ်လို့ရပါမည်။ ပုံ (၄.၂) က Ex401.cpp program ကို run လိုက်လို့ရပါမည့် output result ပြန်ပါသည်။



ပုံ (၄.၂)

Using the Pointer in a Loop

၁။ အောက်ပြုမည့် Ex402.cpp program ကတိုရ် for loop ထဲမှာ pointer ကိုအသုံးပြုပြီး 10 ကနေ 1 ကို ရွတ်ဖတ်ကာမှောင်ခိုင်းပါလိမ့်မည်။ ပြုရာ BOOM! ဆိုတဲ့စကားပေး သက်သေကာမှောင် ပုံ (၄.၃) မှာပေးထားတဲ့ program က code statement ဆွဲလို့အသုံးပြုကြည့်ပါ။

Ex402.cpp program ကို trace လုပ်ပြသပါမည်ရေး

- program ပထမ countDown ဆိုတဲ့ variable ကို type int လို့ပေးကြပါသည်။ ပြီးနောက် pointer cpd ကို define လုပ်ပြီး countDown နဲ့ address value နဲ့ assign လုပ်ပေး

```

Ex402.cpp
// Listing 4.2 Using a pointer, this program
// counts from 10 to 1 displaying the numbers
// followed by "BOOM!".

#include <iostream>

int main()
{
    int countDown;
    int *p = &countDown;

    for (countDown=10, countDown--, countDown = 1;
        cout << *p << ", ";
        cout << "BOOM!" << endl;
        return 0;
}

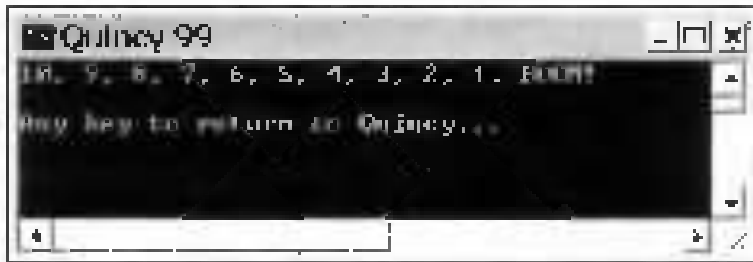
```

Figure 4.2

ဤ for loop သည် countDown ကို 10 သို့ 1 သို့ decrement လုပ်ပြီးနောက် *p ကို display လုပ်ပေးမည်။

- သို့မဟုတ် *p ပေါ်တွင် countDown = 10 သို့မဟုတ် *p ကို display လုပ်ရာတွင် 10 ကို ပြသပေးမည်။ ဤကဲ့သို့ comma (,) ကို အသုံးပြုပေးခြင်းဖြင့် for loop ကို သင်္ချာပြန်လှည့်ပေးမည်။ countDown = 9 ဖြစ်သည့်အခါ သို့မဟုတ် *p ပေါ်တွင် countDown = 9 ဖြစ်သည့်အခါ *p ကို 9 ဖြစ်သည့်အခါ *p ကို display လုပ်ပေးမည်။ 9 သို့မဟုတ် 9။
- ဒုတိယအကြိမ် for loop သည် countDown = 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, သို့မဟုတ် display လုပ်ပေးမည်။ for loop ကို အသုံးပြုပေးမည်။ cout << "BOOM!" << endl; statement သို့မဟုတ် 1 ကို ပြသပေးမည်။ BOOM! ကို display လုပ်ပေးမည်။

Ex402.cpp program ကို Figure 4.2 မှာ run ပြုနိုင်မည်။ Output result သည် အောက်ဖော်ပြပါအတိုင်းဖြစ်သည်။



ပုံ (၄.၄)

၄.၂ Pointers and Arrays

၁။ နှိုင်း variable ဆွေးနွေးရာတွင် address ဆက်ဖက်တဲ့အခါမှာ & (ampersand) ကသည့်အခါဆိုရင်အောက် array ဆွေးနွေးရာတွင် array name တို့အားကိုးကာ address ခြုံနှစ်နေတာကို & ကျင့်ပြီးအစီအစဉ် array notation pointer notation နဲ့ pointer ကသည့်ပြင်အစွဲတဲ့ ပုံ (၄-၅) က Ex403.cpp program မှာပိုင်အသုံးပြုနိုင်ကြောင်း ပြထားပါ။ ရေးသားကြည့်ရအောင်။

၂။ Ex403.cpp program ကို trace ဖတ်ကြည့်ရမယ်ဆိုရင်

- အစမှစပြီး type int (ဖြစ်ပေ) array အင်မတန် define ဖတ်သလို initialize ဖတ်ပြီးနောက်မှာပင် array name ကို x ဟု x အနက်အဓိပ္ပာယ်ကို array element အင်မတန် array notation အသုံးပြုပြီး အစမှစပြီး display ဖတ်နိုင်ပေအောင် $x[0] = 10$ ၊ $x[1] = 20$ ၊ $x[2] = 30$ ၊ $x[3] = 40$ နှင့် $x[4] = 50$ ဆိုပြီးဖြစ်ကြောင်းကို တွေ့ရပါမည်။
- ပုဒ်အစမှစပြီးနောက်မှာပင်အောက် array element များကို pointer notation အသုံးပြုပြီးနောက် display ဖတ်နိုင်ပေအောင် $*(x+1)$ ဟု $x[1]$ နှင့်အတူတူပင် $x+2$ ဟုဆိုရင် x array ရဲ့အောက်မှာ ဖြစ်ပေ integer နဲ့ address ကိုဆိုလိုပေအောင် $*(x+2)$ ဟု 30 ကိုဆိုရင်ဆိုပြီးနောက်မှာပါ။
- address ဖွင့်ကြည့်ခြင်းများအားဖြင့် x မှ $x + 1$ ကသည့်အခါမှာပင် pointer မှ increment ကို အသုံးပြု ဖြစ်နိုင်ပေအောင် x မှ $x + 1$ ကို point ဖတ်ရမှာကို pointer ကြားကသည့်အခါမှာ $*(x++)$ ကို loop ထဲမှာ display ဖတ်နိုင်ပေအောင် array element ကိုအသုံးပြုအောင် ကာယကုန်ပါ။

၃။ Ex403.cpp ကို ပုံ (၄-၆) မှာ run ကြည့်ရပါမည်။ program နှိုင်းအစိုး ရေးသားကြည့်ရန်အတွက်ပေးပါမည်။

```
File403.cpp

// Listing 4.3. This program shows the different
// ways how the arrays are accessed.

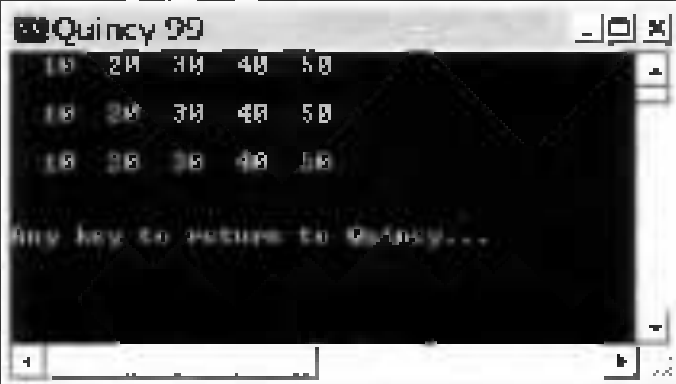
#include <iostream>
int x[5] = {10,20,30,40,50};

int main()
{
    // array accessed with array notation
    for (int i=0; i<5; i++)
        cout << " " << x[i];
    cout << "\n\n";

    // array accessed with pointer notation
    for (int i=0; i<5; i++)
        cout << " " << *(x+i);
    cout << "\n\n";

    // array accessed with pointer
    int * xp = x;
    for (int i=0; i<5; i++)
        cout << " " << *(xp+i);
    cout << "\n\n";
    return 0;
}
```

❖ (5-5)



```
Quincy 99
10 20 30 40 50
10 20 30 40 50
10 20 30 40 50
Any key to return to Quincy...
```

❖ (5-6)

Pointer Arithmetic

Ex 4.4: Ex404.cpp program uses pointer notation to store array values, average program displays program code statement

```
Ex404.cpp
// Listing 4.4: This program reads five numbers that
// you type in, and store them in an array. The program
// would average them and display the result.

#include <iostream>

using namespace std;

const int MAX = 5;
char ch;
int k=0;
float f[MAX];

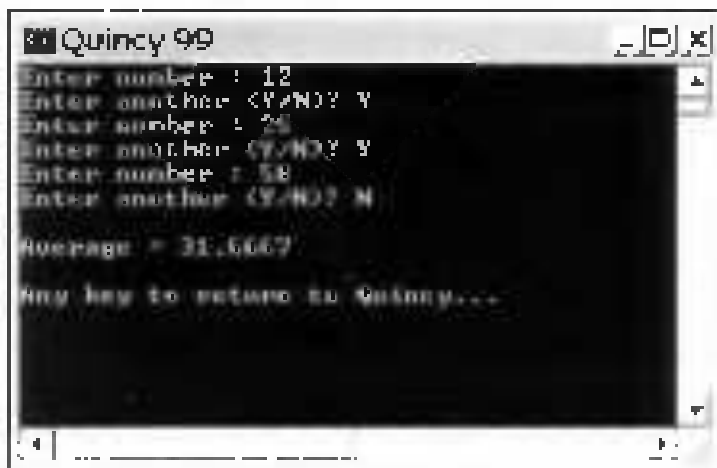
int main()
{
    for (int i=0; i<MAX; i++)
    {
        cout << "Enter number: ";
        cin >> f[i];
        cout << "Enter another?(Y/N)? ";
        ch = cin.get();
        if (ch == 'N') break;
    }

    float sum = 0;
    for (int i=0; i<k; i++)
        sum += f[i];
    float avg = sum/k;
    cout << endl << "Average = " << avg << endl;
    return 0;
}
```

(3-2)

Ex404.cpp program: တို့အသုံးပြုပုံအသစ်ဆိုရင်

- for loop & pointer notation ဖြစ်ပြီး k array မှ, element တစ်ခုခု၏တန်ဖိုး value ကို x နှိုက်ကည့်အသွယ်ကလေး `cin >> *(x - k++);` statement ဖြစ်ပြီး k က ကျွန်ုပ်တို့ input လုပ်မည့် array element တစ်ခုခုကိုဖော်ပြသော variable တစ်ခုဖြစ်ပါသည်။ data ကိုဆက်လက်သွင်းရန် အသစ်ဆိုရင် 'Y' ကိုနှိပ်ကည့်ပါ။ data တစ်ခုခုခုရင်လည်း 'N' ကိုနှိပ်ကည့်ပါ။ ဒါဆိုရင် break နှင့်အတူပဲ for loop ကြိုတင် တိုင်းပုံကွက်သတ်နိုင်မည်။
- အခုဆိုရင် count ဆရာ တစ်ကယ်အဆိုရှိတဲ့ array element အရေအတွက်ကလေး array element မှနေပါင်းခံ့တစ်ခုကို ပေါင်းခံ့ပါပြီးကော sum ဆရာ store လုပ်ပေးပါသည်။ မနက်ပြီး `avg = sum/k` နှင့်ကွက်ပျဉ်း အဖွဲ့ကို display လုပ်ပြလိုက်ပါပြီ။ ပုံ (၄. ၈) မှ Ex405.cpp program ကို run လိုက်ကြည့်ရမည် result ကိုပေါ်ပြထားပါသည်။



ပုံ (၄. ၈)

- ထပ်မံပေးသည့် pointer ကို နှိုက်နှိုက်အသုံးပြုပြီး Ex404.cpp program ကိုဖြင့် အမှတ်ဆိုရင် ပုံ (၄. ၉) မှလိုသကဲ့သို့ program ကို run ကြည့်ရမည်။ ဒီ program ကို run မှုလုပ်ပြီး မေးခွန်းက အဖြေခံရမည်။

```

Ex004A.cpp
#include <iostream>

const int MAX = 100;
char ch;
int i = 0;
float f[MAX];

int main()
{
    float xp = 0;

    for (int i=0; i<MAX; i++)
    {
        cout << "Enter number "
              << i << " : ";
        cin >> f[i];
        cout << "Enter another float : ";
        cin >> ch;
        if (ch != '\n') break;
    }

    float sum = 0;
    for (int i=0; i<MAX; i++)
        sum += f[i];

    float avg = sum/i;
    cout << endl << "Average = " << avg << endl;
    return 0;
}

```

Figure 9.2

9.2 Pointers to Structures

• structure \rightarrow pointer \rightarrow pointer \rightarrow pointer \rightarrow pointer \rightarrow pointer
 structure pointer \rightarrow structure type \rightarrow instance \rightarrow pointer \rightarrow pointer
 \rightarrow increment \rightarrow decrement \rightarrow structure size \rightarrow pointer \rightarrow pointer
 \rightarrow structure \rightarrow member \rightarrow pointer \rightarrow access \rightarrow member pointer

operator (->) ကို အသုံးပြုရန် နှင့် (&) နှင့် ဖော်ပြထားတဲ့ Ex405.cpp program ကို အသုံးပြု point-
ers to structures ဆိုတဲ့အခန်းကို ဆက်လက်အားဖြည့်ထားတဲ့ program ကို ဖော်ပြပါ။

```
Ex405.cpp
// Listing 4.5 Pointers to structures
#include <iostream>

struct employeeRec
{
    int    emplNo;
    float  hours;
    float  wage;
};

void displayRec(const employeeRec *y;
{
    cout << "\nEmplNo Hours Wage Pay\n";
    while (y->emplNo != -1)
    {
        cout << y->emplNo << " ";
        cout << y->hours << " ";
        cout << y->wage << " ";
        cout << (y->hours)*(y->wage) << endl;
        y++;
    }
}

int main()
{
    employeeRec a[] = {
        {1, 65, 30},
        {2, 50, 50},
        {3, 40, 75},
        {-1, 0, 0}
    };

    displayRec(a);
    return 0;
}
```

↳ (g. 00)

၁ Ex405.cpp program သို့မဟုတ် ကြည့်ရှုရန်အတွက်

- နောက်ဆုံး employeeRec သို့မဟုတ် structure သစ်ကို declare လုပ်ထားသည်။ main() function ထဲမှ employee instance တစ်ခုဖြစ်တဲ့ x[] array ကို declare ကြိုတင်ထားပြီး နောက်ဆုံး employeeRec structure variable ကို initialize လုပ်ထားသည်။ ထိုသို့မဟုတ် ထပ်မံ displayRec() function ကို call လုပ်ပြီး x[0] element ရဲ့ address ကို pass လုပ်ထားသည်။ ထိုသို့မဟုတ် displayRec() function ထဲမှ နောက်ဆုံး array Element ကိုရတဲ့ ယူကော့ကို မှတ်တမ်းတင်ပေးနိုင်မယ့် တွက်ချက်မှုနှင့် pointer y ကို increment လုပ်ခြင်း အတိုင်း x[1] element + x[2] ကိုမှတ်တမ်းတင်ပေးပြီး y &empId = 1 ခြင်းအားဖြင့် ထပ်မံ displayRec() ကို ခေါ်ဆိုလိုက်တဲ့ program ကြည့်ရှုရန် နံပါတ် ၁၀ မှာ Ex405.cpp program ကို run ခြင်းဖြစ်သည်။



ပုံ ၄.၅ (၁)

၄.၄ Pointers as Function Arguments

၁။ program ကိုမူ့ argument ကို function ထဲမှ ခေါ်ဆိုပေးရန် pass လုပ်ခြင်း (၁) ရှင်းရှင်းလင်းလင်း (၂) passing argument by value (၃) passing arguments by reference နှင့် (၄) passing argument by pointer သို့မဟုတ် pointer ကို အသုံးပြုခြင်းတို့ကို အသုံးပြုခြင်းဖြစ်သည်။ (၅) ခ။ Ex406.cpp မှ ကြည့်ရှုရန် passing by reference ရှင်းရှင်း တစ်ခုကို အသုံးပြုခြင်းဖြစ်သည်။

```

Ex406.cpp
printing 4.0 This program shows how arguments
are passed by reference
#include <iostream>

void convert(double &);

int main()
{
    double inches;

    cout << "Enter inches: ";
    cin >> inches;
    convert(inches);
    cout << "Length = " << inches << " centimeters\n";
    return 0;
}

void convert(double &i)
{
    // convert inches to centimeters
    i *= 2.54;
}

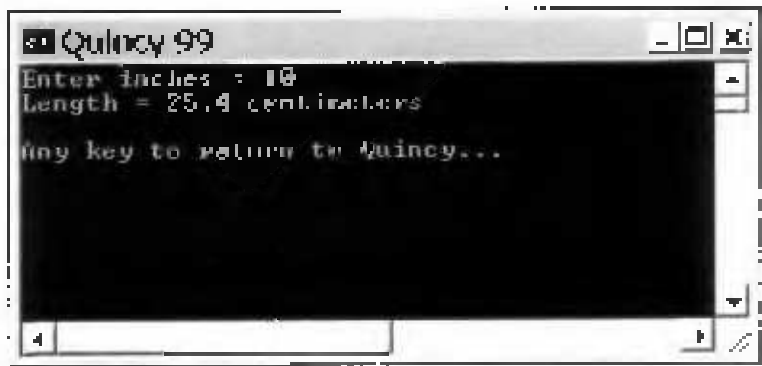
```

Figure 10.10

Ex406.cpp program showing pass-by-reference

- main() function uses inches as a variable and pass its address to convert() function. convert() function converts inches to centimeters and returns the result to main() function by reference. In this example, data pass is done by reference. In main() function, convert() function is called with inches as an argument and pass its address to convert() function. In this example, data type is double and data pass is done by reference. In this example, data pass is done by reference and the program will output the result of the conversion and return the result to main() function.

- Ex406.cpp program ကို run လိုက်ပြီး inches = 10 ကိုရိုက်ပါသည်။ Length = 25.4 centimeters နည်းပေးခြင်းပဲ (၄-၁၄) မှာပြထားတဲ့ကတည်း၊ ပြင်ပေါက်ပုံပေါင်း



ပုံ (၄-၁၄)

- Ex406.cpp program ကို passing by pointer နည်းနဲ့ပြင်ရေးမယ်ဆိုရင် ပုံ (၄-၁၄) မှာပြထားတဲ့ program မျိုးကိုတွေ့ရမှာပေါ့။ ဒီ program ကို run ရင်လည်း တေးကေးကေးပြင်ရမှာပေါ့။

```

Ex406A.cpp
#include <iostream>
using namespace std;
void convert(double* xp)
{
    // Convert inches to centimeters
    *xp *= 2.54;
}

int main()
{
    double inches;
    cout << "Enter inches: ";
    cin >> inches;
    convert(&inches);
    cout << "Length = " << inches << " centimeters\n";
    return 0;
}

```

ပုံ (၄-၁၅)

Passing Arrays by Pointers

- Function arguments array are pass သုံးပုံစံတို့ထဲတွင် အများဆုံးဖြစ်ပြီး array notation မှ pass သုံးပုံစံထက် pointer မှည့်ပေး အသုံးပြုပေးရပါသည်။ (၄-၁၅) on Ex407.cpp program မှ array passed by pointer မှည့်ပေးပါကြည့်ပါအောင် ၊ အောက်ပြန်ညှိပါ။

```
Ex407.cpp
// Listing 4.7 The program shows how arrays are
// passed by pointer.

#include <iostream>
const int MAX = 5;
void convert(double*);

int main()
{
    double avg, sum = 0;
    double inches[MAX] = {10, 20, 30, 40, 50};

    convert(inches);
    for (int i = 0; i < MAX; i++)
    {
        cout << endl << "Length[" << i + 1 << "] = " << inches[i] << " cm";
        sum += inches[i];
    }
    avg = sum / MAX;
    cout << "\n\n" << "Average = " << avg << " cm";
    cout << endl;
    return 0;
}

void convert(double* xp)
{
    for (int i = 0; i < MAX; i++)
        *xp++ *= 2.54;
}

```

မှ (၄-၁၅)

Ex407.cpp program ကိုရေးသားကြည့်ပါ။

- main() function ကို type double variable (2) ခြင်း၊ avg နဲ့ sum တို့ကို define ခုတ်၊ sum = 0 နဲ့ initialize ခုတ်၊ inches နာမည် array ကို define + initialize ခုတ်၊ inches array element အရေအတွက် 5 နဲ့ သတ်မှတ်ပါ။ inches[0] = 10, inches[1] = 20, inches[2] = 30 သတ်မှတ်တဲ့ ခြင်းပါ။
- convert() function ကို call ခုတ်၊ inches array နဲ့ address ခြင်း၊ array name နှင့် address - ခြင်း၊ convert (inches) ကို pass ခုတ်၊ inches array ကို pointer to double ကို ရောက်စေရန်၊
- for loop ကို ခုတ်၊ xp ကို point ခုတ်၊ inches address နာမည် element ကို ရောက်စေရန်၊ နာမည် array element ကို ရောက်စေရန်၊ array element ကို cout နဲ့ 0 element နာမည် ခြင်း၊ pointer xp ကို indirection operator ကို *xp++ ကို increment ခုတ်၊ inches array ကို ရောက်စေရန်၊ main() ကို inches element နာမည် array element ကို ရောက်စေရန်၊
- main() function ကို inches array element ကို ရောက်စေရန် display ခုတ်၊ sum + inches[i] ကို ရောက်စေရန် array element ကို ရောက်စေရန်၊ sum နဲ့ assign ခုတ်၊ inches array element ကို ရောက်စေရန်၊ display ခုတ်၊ inches[i] ကို ရောက်စေရန်၊ program ကို ရောက်စေရန် output result ကို ရောက်စေရန်။



ပုံ (၂.၁၆)

Swapping Two Numbers with Pointers

ဤပုံစံသည် pointer သုံးခုဖြင့် array သို့ access သုံးခုဖြင့် ပိတ်ထားပြီးရှိရုံဖြင့် ကိန်းတို့ကို ပြောင်းလဲပေးရာ array element တို့ကိုဖွဲ့ခြင်းကို လက်တွေ့စမ်းကြည့်လိုရမည့် အလွယ်တကူပေးပါသည်။ (2) မှ ဤ function သုံးခုကို pass သုံးခုဖြင့် pointer သုံးခုဖြင့် လုပ်ငန်းဆောင်ရွက်ကြပါသည်။ (3) မှ (4) မှာ ဤ program ကိုလုပ်ကြည့်ပါ။

```
Ex108.cpp

// Listing 4.8 This program swaps two numbers
// passed to function as argument.

#include <iostream>
using namespace std;

int main()
{
    int first=95, second=12;
    cout << "Before swapping";
    cout << "\nFirst = " << first;
    cout << "\nSecond = " << second;

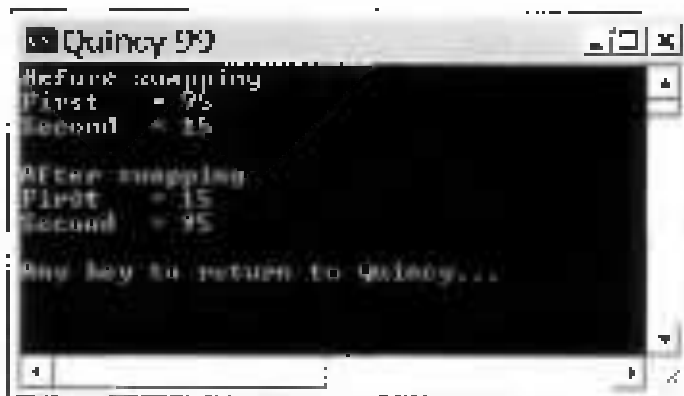
    swap(&first, &second);
    cout << "\nAfter swapping";
    cout << "\nFirst = " << first;
    cout << "\nSecond = " << second << endl;
    return 0;
}

void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

```

၂ Ex408.cpp program ကိုလေ့လာကြည့်ဆင်ယင်ဆွဲဆောင်

- မူလအခြေအနေ first = 95၊ second = 15 သို့ integer variable (2) နှစ်ကို initialize လုပ်ပေးထားပြီး swap လုပ်ဆောင်မှာပင် (2) နှစ်ကို display လုပ်ပေးပြီးနောက် swap() function ကို call ပြုပေးထားပြီး passing argument ဆောင်ရာ first နှင့် second ကိုလည်း address လုပ်ပေးထားသည်။
- ၅ address မှလွှဲထား pointers to integers first နှင့် second ဖြစ်လာပြီး x နှင့် y ကိုလည်း pass လုပ်ဆောင်ဆောင်ရာ swap() function ဆဲမှာ *x နှင့် *y ကိုလည်းလုပ်ဆောင်ပေးပေးထားပြီး ဖြစ်ပြီး *x = 15 နှင့် *y = 95 ဖြစ်လာပြီး main() function ကိုပြန်လှောက်လှည့်ဆောင်ရာ ၅ non-break (2) နှစ်ကို display လုပ်ပေးခြင်းဖြစ်သည်။
- Ex408.cpp program ကိုလည်းကောင်း output result ကိုလည်း (၄. ၁၅) ဖြစ်လာပေးပေးနိုင်ခြင်းဖြစ်သည်။



```
Quincey 99
Before swapping
First = 95
Second = 15

After swapping
First = 15
Second = 95

Any key to return to Quincey...
```

ပုံ (၄. ၁၅)

Bubble Sort

၁။ ပုံ (၄. ၁၉) ဖြစ်လာပုံ Ex409.cpp program မှာဆွဲဆောင် array ကနေဆွဲဆောင် element ဆွဲဆောင်ပေးပြီးလျှင် pointer မှလွှဲပြီး bubble sort ဆောင်ရာမှာ program မှလွှဲဆောင်ပြီးလျှင် trace လုပ်ကြည့်ဆင်ယင်ဆွဲဆောင်

```

// Listing 4.9: This program sorts an array
// using the bubble sort method.

#include <iostream>
using namespace std;
void swap(int*, int*);
void bubbleSort(int*, int);

int main()
{
    const int N=10;
    int x[N] = {17, 50, 25, 70, 45, 30, 95, 20, 60, 100};

    cout << "Before sorting\n";
    for(int j=0; j<N; j++)
        cout << x[j] << " ";
    bubbleSort(x, N);
    cout << "\nAfter sorting\n";
    for(int j=0; j<N; j++)
        cout << x[j] << " ";
    return 0;
}

void swap(int* a, int* b)
{
    int* temp = a;
    *a = *b;
    *b = *temp;
}

void bubbleSort(int* a, int n)
{
    for(int j=0; j<(n-1); j++)
        for(int k=j+1; k<n; k++)
            swap(a[j], a[k]);
}

```

Ex409.cpp program တွဲလျှောက်ကြည့်မယ်ဆိုရင်

- bubbleSort() function ထဲမှာ inner loop ပတ်လုံးကုန်း ပထမကကန်း (10) နဲ့ ဒုတိယကကန်း (50) ကိုဖွဲ့ပြီးလျှင်ကြည့်ပါလား။ ဒုတိယကကန်းက ပထမကကန်းထက်ငယ်လာမီဆိုရင် မှန်ရာသွင်းထဲ ပါမယ်။ မကပ်ဘူးဆိုရင် ပထမကကန်းနဲ့ ထက်ပေး>ကန်း (20) နဲ့ယှက်လျှင်ကြည့်မယ်။ ထက်ပေးကကန်းက ပထမကကန်းထက်ပေးမယ် ဟုဆိုရမယ်။ မှန်ရာသွင်းပထမပေးပေးပါလား။ ဒီနည်းကတိုင်း ယက်လုပ်သူမခင် outer loop အတွက် ပထမပေးကြိမ်ပြုသွင်းပါပြီ။ ပထမပေးကြိမ်ပြုတဲ့ကခါမှာ *[] array ကအမှန်ဖြစ်နေပါလိမ့်မယ်။ အမှန်ပြောင်းထဲမှမဆိုသေးပါဘူး။

10, 50, 20, 70, 40, 30, 90, 80, 60, 100

- outer loop ဒုတိယကကြိမ်စရင် *[] array ရဲ့ဝယ်ထက်အစွန်ဆုံးကန်းပေး။ ဖလှယ်မဲ့ဖြစ်နေပါလိမ့်မယ်။ ဒုတိယကန်းကို မှန်ပေးပါလား။ အမှန်ပေးပါမယ်။ ပထမကကန်းကို (50) ကိုယက်မှတ်ရပါမယ်။ ကွန်ထရိုကန်း (9) လုံးထဲက ဝယ်ယူခြင်းကို (20) ကို ဒုတိယပေးအစွန်ဆုံးမှာ မှောက်တာပင် စာစာစာကတိုင်း swap လုပ်ခဲ့ရင် *[] array ကအမှန်ဖြစ်နေပါလိမ့်မယ်။

10, 20, 50, 70, 40, 30, 90, 80, 60, 100

- ဒီနည်းကတိုင်းဆက်လုပ်သူမရင် *[] array element မလွှဲဟာ အမှန်ပင် တစ်ဆင့်ပြီးတစ်ဆင့် မှန်ရာသွင်းသွင်း sorting ဟုပြောသွားပါလိမ့်မယ်။

10, 20, 30, 50, 70, 40, 90, 80, 60, 100

10, 20, 30, 40, 50, 70, 90, 80, 60, 100

10, 20, 30, 40, 50, 60, 70, 90, 80, 100

10, 20, 30, 40, 50, 60, 70, 80, 90, 100

- Ex409.cpp program ကို run လိုက်ရင် output result ကို ဝို (၃, ၈၈) ဟုပြောထုတ်ကတိုင်း ဖြစ်ပေါ်လိမ့်မယ်။



ပုံ ၂.၄.၁၀

၄.၆ Pointers and Strings

String ဆိုသည်မှာ variable type array သမိုင်းအစဉ်အလာဖြစ်ပါသည်။ ဒီအတိုင်း array element တို့ကို pointer notation နှင့် access ရန်လိုအပ်ပြီး string array ကိုလည်းကောင်း၊ string array ကိုလည်းကောင်း increment ရန်လိုအပ်သည်။ pointer ဆိုသည်မှာ မှည့်ဆိုရန်ပါသည်။ ဥပမာ (ပုံ ၂.၄.၁၀) မှာ Ex4010.cpp program မှာလိုအပ်ပြီး pointer notation ကိုအသုံးပြုပြီး အစဉ်အလာပြပါသည်။

Ex4010.cpp program ကိုအသုံးပြုပြီးပါကလိုအပ်

- str နှင့် strp (pointer to str) ကို define ရန်နှင့် initialize ဆောင်ရွက်ပါသည်။ ဤအတိုင်း str နှင့် strp ကို display ရန်အတွက်အသုံးပြုပါသည်။
- for loop တွင် str ကို increment ဆောင်ရွက်ပြီးနောက် str ဆိုသည်မှာ character (၅) ရှိရာ pointer variable မှာ store ဖြစ်သွားပါမည်။ strp ကို display ရန်အတွက် အသုံးပြုပါသည်။

```

Ex4010.cpp

// Listing 4.10 This program shows how the strings
// can be defined using array and pointer notations

#include <iostream>

int main()
{
    char str[] = "Defined as an array";
    char* strp = "Defined as a pointer";

    cout << endl << str << endl << strp;

    for (int i=0; i<8; i++)
        strp[i];
    cout << endl << strp << endl;
    return 0;
}

```

ပုံ (၄-၂၀)

- Ex4010.cpp program ကို run လိုက်ရင် output result တို့ ပုံ (၄-၂၁) မှာပြသထားတဲ့အတိုင်း ပြင်ဆင်နိုင်ပေမယ့် str string ထဲက character (8) လုံးပေါ်လာတာကိုတွေ့မှာပဲ။

```

Quincy 99
Defined as an array
Defined as a pointer
as a pointer

Any key to return to Quincy...

```

ပုံ (၄-၂၁)

Changing Characters of an Array to Uppercase

Figure 9-19 shows the Ex4011.cpp program that character array တစ်ခုထဲမှာ element များကို uppercase letter ဆွဲချက်ဖြစ်အောင် pointer ဆက်သွယ်ခြင်းဖြင့်ပေးတဲ့ program ဖြစ်ပါတယ်။ Figure 9-20 မှာ Ex4011.cpp program ကို run ဖြောဟပါတယ် ၊ ဆရာတော်ကြည့်ပါ။

```
Ex4011.cpp
// Listing 9-11 This program requests a string from
// the user and then displays the string in uppercase letters
#include <iostream>
#include <ctype>

void copyUpper(char* srcp)
{
    while(*srcp) *srcp++ = toupper(*srcp);
}

int main()
{
    char str[80];

    cout << "Enter a string\n";
    cin.getstr(str);
    copyUpper(str);
    cout << endl << str << endl;
    return 0;
}
```

Figure 9-19

```
Quincy 99
Enter a string
Pointers and Addresses
POINTERS ARE FUN TO USE.
Any key to return to Quincy...
```

Figure 9-20

Strings as Function Arguments

Ex 4012: Ex4012.cpp program uses string as a function argument passing by pointer. The program displays each character in turn.

```
Ex4012.cpp

// Using A.12 This program shows a string
// used as a function argument. The function
// displays the string each character in turn.

#include <iostream>
using namespace std;

int main()
{
    char str[] = "Display a string";
    char *ptr = "INCREMENT BY ONE";

    displayStr(str);
    displayStr(ptr);
    return 0;
}

void displayStr(char *str)
{
    while (*str) cout << *str++ << endl;
}
```

(A.12)

Ex4012.cpp program is run. The output result shows string as a pointer is pass as a pointer of increment of a string character display.



Figure 1.10

Copying a String Using Pointers

Figure 1.11 shows the Ex4013.cpp program and the main() function. The copyStr() function is called with a string and a pointer array as arguments. The copyStr()

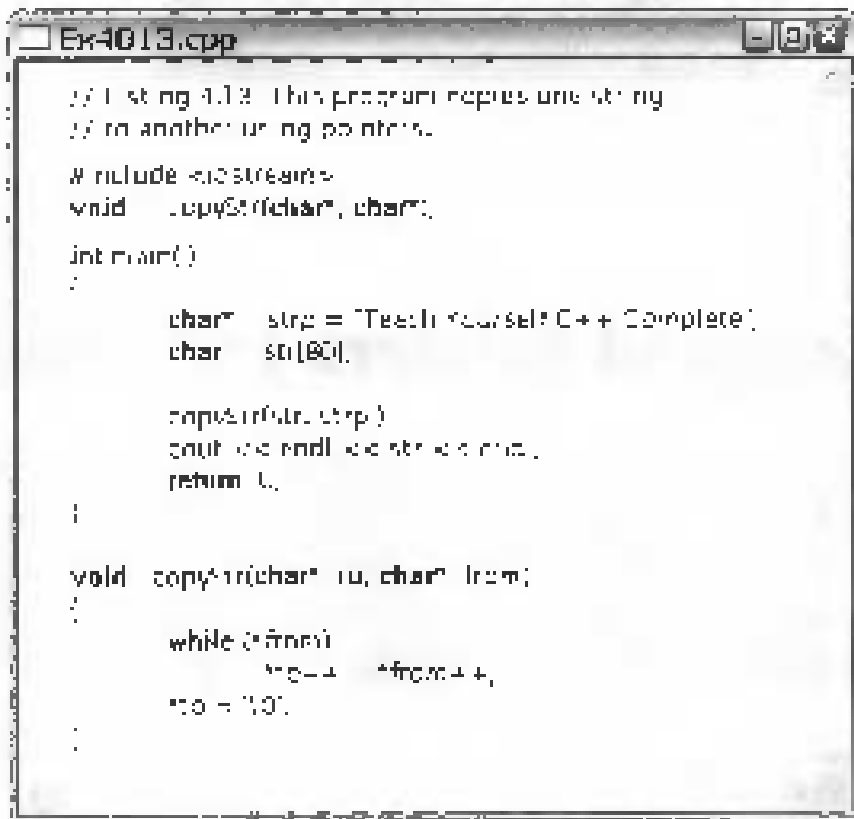


Figure 1.11


```

Ex4014.cpp
// Listing 4.14 This program creates an array of pointers
// to strings, rather than an array of strings,
// to another using pointers.

#include <iostream>
const int MAX = 7;

int main()
{
    char* strp[MAX] = {"TIME", "IS", "A GREAT",
                      "TEACHER,AN", "BUT IT",
                      "KILLS ITS PUPILS."};

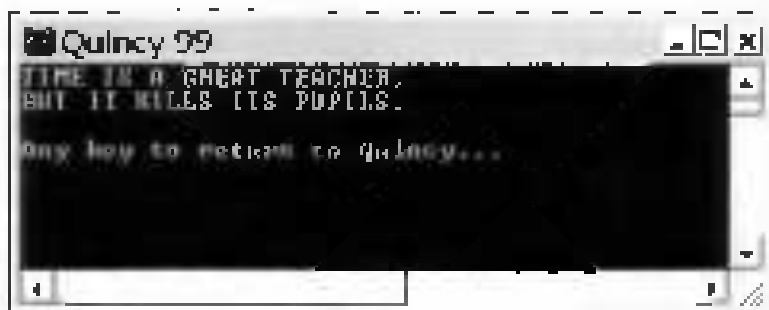
    for (int i=0; i<MAX; i++)
        cout << strp[i]

    cout << endl;
    return 0;
}

```

◆ (၄.၂)

၂။ Ex1014.cpp program ကို run လုပ်ပါကအောက်ဖော်ပြပါနှင့် `strp[i]` pointer array အရ pointer element အစဉ်အတိုင်း point လုပ်ရာတွ် string အစဉ်အတိုင်းအတိုင်း နှိုင်း array အစဉ်၊ element အစဉ် display လုပ်နေကြောင်းကို display လုပ်ကြည့်ရမည်။ (၄.၂) ကိုကြည့်ပါ။



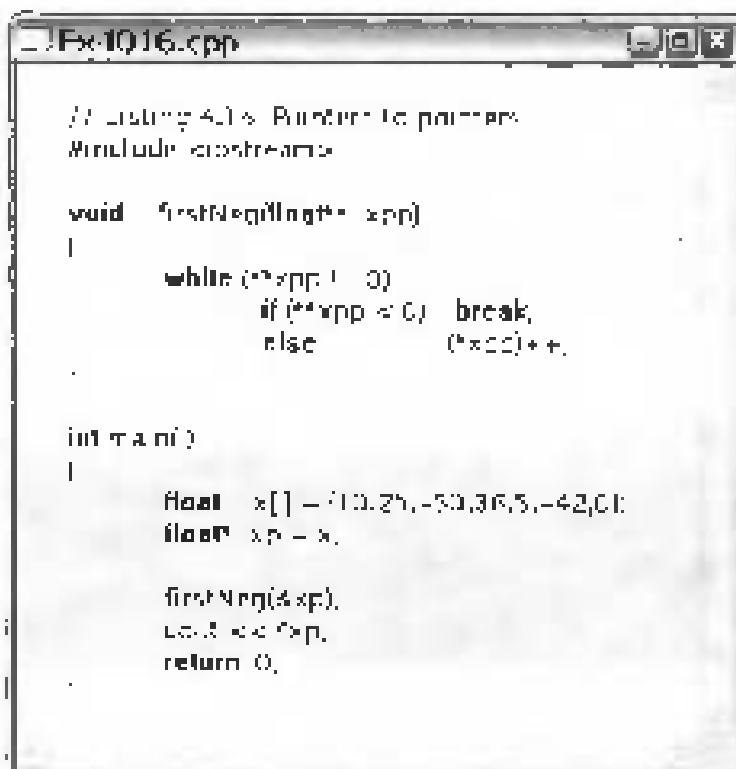
◆ (၄.၂)

၄.၈ Pointers to Pointers

pointer ကနေ pointer ကို point လုပ်လေ့ရှိရင် * (asterisk) (?) ရဲ့ တွက်စနစ်ပြုရပါမယ်။ ဥပမာ CD ၁၀၀ char pointer ခေါ်ဆိုရင် char** cp = &cp, သို့မဟုတ်ဆိုရပါမလား။ cp က pointer to pointer ဖြစ်လာမယ်၊ pointer to pointer ကို initialize လုပ်လေ့ရှိရင် အမှတ်ပေးရပါမလား။

```
char c = 'A', // char variable
char* cp = &c; // pointer to char variable
char** cpp = &cp; // pointer to pointer
```

pointer to pointer ကနေ point လုပ်နေတဲ့ pointer ကိုမှည့်တဲ့ char variable ဆိုတာ data ကို ဆိုလျက်ပေးရင် char*cp1 = *cpp; သို့မဟုတ် char c1 = **cpp; လို့ရေးပြီး retrieve လုပ်ရင်ပေးမယ်။ ဆောင်း၍ ၆ (၄...၅) မှာရေးထားတဲ့ Ex4016.cpp program ကိုအမှတ်ပြုပြီး pointer to pointer ကို အသေးစိတ်ကိုင်စီမံမယ်။



```
Ex4016.cpp
// Listing 4.16: Pointer to pointers
#include <iostream>

void firstNeg(int** xpp)
{
    while (*xpp != 0)
        if (**xpp < 0) break;
        else (**xpp)++;
}

int main()
{
    float x[] = {10,25,-30,36,5,-42,0};
    float* xp = x;

    firstNeg(&xp);
    cout << xp;
    return 0;
}
```

၆ (၄-၂၆)

Ex4017.cpp program array of pointer to pointer & access of array element

```

Ex4017.cpp
// Listing 4.17 Pointers to arrays of pointers
// An array of pointers

char str[] = {"Zara",
             "Subu",
             "Sungjune",
             "Yehliwei",
             "Feng",
             0};

int main()
{
    char** pp = str;

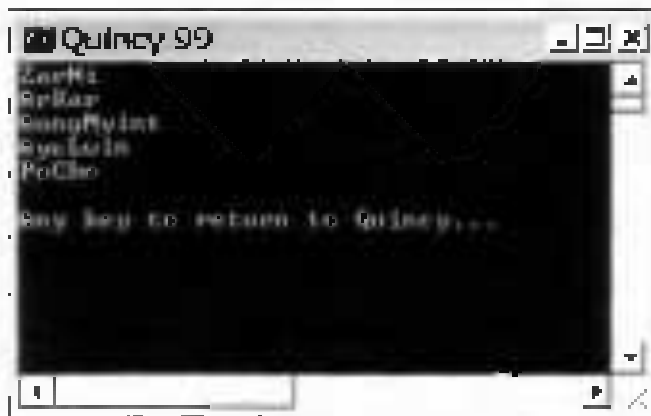
    while (*pp != 0)
        cout << *pp++ << endl;

    return 0;
}
    
```

Ex 4.17

Ex4017.cpp program array of pointer to pointer

- array of char pointer array & initialize with string point array
 & first element of null pointer of array is 0. main() function array
 pointer to pointer pp of pointer array & assign with while loop
 *pp of pointer array & display of array element *pp++ shows
 next pointer array element & point to next array element
- Ex 4.17: Ex4017.cpp program of run result



ပုံ (၄-၃၂)

Pointers to const Variables

၁၈ ပုံ (၄-၃၄) နမူနာအဖြစ် Ex401B.cpp program မှာ pointer to const variable အသုံးပြုပုံကို ကြည့်ရှုနိုင်သည်။ main() function ထဲမှာ const char variable အဖြစ်ကို str1[] ကို string အဖြစ် initialize လုပ်ပေးပြီး copyUpper() function ကို call ပေါ်ပါသည်။ calling function မှာ str1 နဲ့ str2 char array (2) မှာကို pass လုပ်ပေးပြီး called function ထဲမှာ char pointer အဖြစ်ကို s1 + s2 ကိုတင်သတ်ပါသည်။ function return type ကို char pointer လိုအပ်ပုံအတိုင်းအတာကို သတိပြုကြည့်ပါ။ called function ထဲမှာ s1 ကို point လုပ်ပေးတဲ့ pointer ကို s ကို assign လုပ်ပေးပြီး s2 point လုပ်ပေးတဲ့ str2 ထဲမှာ character အဖြစ်အနေဖြင့် uppercase letter ပြန်ဖော်ပေးပုံအတိုင်း *s1 + ကို assign လုပ်ပေးပါသည်။ အောက်ပါအတိုင်းအတာဖြင့် s1 ကို point လုပ်ပေးတဲ့ pointer s ကို main() ဆီ return လုပ်ပေးပေးပါ။ ဒီလို main() function မှာ str1 ကို display လုပ်ပေးရင် str2 ကို uppercase အဖြစ်အတိုင်းအတာဖြင့် ပြန်ပြန်ရပါမည်။

၂၀ Ex401B.cpp program ကို run လိုက်သည့်အခါမှာ BEFORE... နဲ့ AFTER... ဆိုတဲ့ အခြေအနေ (၅) ဝင်ရောက်ပုံကို ရှိပုံအတိုင်း ပုံ (၄-၃၅) ကိုကြည့်ပါ။

```

Ex4018.cpp
// Listing 4.18: const pointer arguments
#include <iostream>
using namespace std;

char copyUpper(char s1, const char s2)
{
    char *s = s1;

    while (*s2 != '\0' || !isupper(*s2) || !islower(*s2))
        return s;

    *s = *s2;
}

int main()
{
    char str1[30];
    const char str2[] = "Be the first to know C++";

    cout << "BEFORE..." << str1 << endl;

    copyUpper(str1, str2);

    cout << "AFTER..." << str1 << endl;
    return 0;
}

```

Figure 4.18

```

Quincy 99
BEFORE...
Be the first to know C++
AFTER...
BE THE FIRST TO KNOW C++
Any key to return to Quincy...

```

Figure 4.19

9.2 Memory Allocation

* ಸ್ಥಳೀಯವಲ್ಲದೆ, global storage memory ನ್ನು free store ಅಥವಾ heap ಉಪಯೋಗಿಸಿ heap ವಿಧ: program memory ಅನ್ನು (ಇಲ್ಲಿ) RAM ಉಪಯೋಗಿಸಿ ಪರಿಹರಿಸಿ program ಅರಿತಾಗಲೂ memory allocation ನ್ನು ಅಭ್ಯಾಸಿಸುತ್ತೇವೆ new ನ್ನು operator ನ್ನು ಅಭ್ಯಾಸಿಸುವೆ

```
EX9020.cpp

// Listing 9.20 The new and delete operators
// C++ with C++11

// Declare a data structure
struct Date
{
    int month;
    int day;
    int year;
};

int main()
{
    // Allocate memory for a data structure
    Date* birthday = new Date;

    // Assign values to the structure members
    birthday->month = 10;
    birthday->day = 25;
    birthday->year = 1997;

    // Display the data structure
    cout << "was born on "
         << birthday->month << "/"
         << birthday->day << "/"
         << birthday->year << endl;

    // Return the allocated memory to the heap
    delete birthday;
    return 0;
}
```

9 (9-20)

ဆိုပုံ heap memory ထဲမှ program ထဲမှ memory block သို့မဟုတ် assign လုပ်ထားပါသည်။
 program မှ: မှားတဲ့ block ဆိုတဲ့ heap မှ ပြန်လုပ်ပေးရန် delete operator ကို အသုံးပြုရပါသည်။
 (၄. ၃၅) မှားကြား Ex4020.cpp program မှ new နှင့် delete operator သုံးစွဲပုံကို ပြသပုံ
 ဖော်ပြပါသည်။

Ex4020.cpp program ကို အသုံးပြုခြင်းအားဖြင့် main() function ထဲမှ နေ့စဉ် birthday
 date ထည့်သွင်း memory block ကို heap ထဲမှ ထုတ်ပေးပါသည်။ ဤနေရာမှ ဝင် date, month, day,
 year value တွေကို assign လုပ်ထားပြီးပါသော birthdaymonth ကို display လုပ်ပုံ birthday
 ထဲမှ month value ကို ထုတ်ပေးပါ။ birthday date ထည့်သွင်း memory allocation ကို ပြန်လုပ်ပေးရန်
 delete birthday; ကို အသုံးပြုပါသည်။ (၄. ၃၆) မှ Ex4020.cpp program ကို run ပြုထားပါသည်။



(၄. ၃၆)

Using new and delete Operators with Arrays

တစ်ခုခုကို birthday date ကို array ထဲမှ ထုတ်ပေးရန် ပြုလုပ်ပေးရန် memory allocation ထဲမှ
 ထုတ်ပေးရပါသည်။ (၄. ၃၇) မှားကြား Ex4021.cpp program မှ Ex4020.cpp program ကို
 ပြင်ဆင်ထားပြီးပါသော array ထဲမှ allocated memory ကို ဖြေပေးရန် delete [] birthday ကို
 ထည့်သွင်းပါသည်။ (၄. ၃၈) မှ program ကို run ရန် အကြောင်းအရာကို ပြသပါသည်။


```
Ex4022.cpp
// Listing 4.22: The new operator and dynamic arrays.
#include <iostream>
#include <cstdlib>

int main()
{
    int size;

    // Get the array size from the user.
    cout << "Enter the array size: ";
    cin >> size;

    // Allocate memory for the array.
    int* arrayX = new int[size];

    // Load the array with random numbers.
    for (int i=0; i<size; i++)
        arrayX[i] = rand();

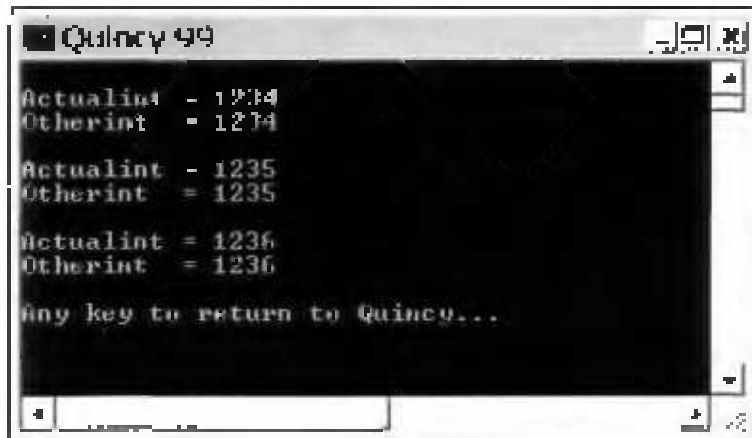
    // Display the array with random numbers.
    for (int i=0; i<size; i++)
        cout << endl << arrayX[i];

    // Return the allocated memory to the heap.
    delete [] arrayX;
    return 0;
}
```

9 (5/28)

```
Quincy 99
Enter the array size: 5
41
18467
6334
26588
19169
Any key to return to Quincy...
```

9 (5/28)



ပုံ (၄.၄၁)

References to Reduce Complex Notation

၁။ C++ program ၏ structure member တွေကို တရားပြန်ဖော်ပြရမယ့်နေရာမှာ reference ကို အသုံးပြုရင် ရွယ်ကောက်ရေးရပါမယ်။ အဲဒီအတွက် Ex4024.cpp program ကို ကျေးဇူးပြု၍ ကောက်ယူရေး ပုံအတိုင်းကိုင် ၊ ဆေးသင်ကြားပုံ။

// Listing 4.24: Initializing a reference

```
#include <iostream>

struct Date
{
    int    month,day,year;
}

struct EmployeeRec
{
    int    empNo;
    char   name[50];
    Date  dates[3];
    float  salary;
}
```

```

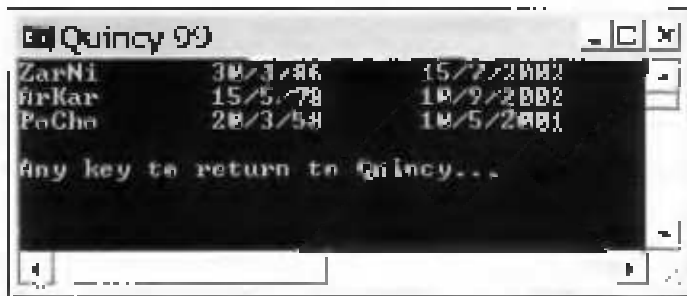
EmployeeRec staff[] =
{
    {001, "ZarNi",{(30,3,86),(15,7,2002)}, 45000},
    {002, "ArKar",{(15,5,78),(10,9,2002)}, 50000},
    {003, "PaCho",{(20,3,58),(10,5,2001)}, 35000},
    {0}
};

int main()
{
    EmployeeRec* xp = staff;

    while (xp->empNo != 0)
    {
        cout << xp->name << " : " << endl;
        for (int i=0; i<2; i++)
        {
            Date& rd = xp->dates[i];
            cout << rd.month << "/" <<
                << rd.day << "/" <<
                << rd.year << " ";
        }
        cout << endl;
        xp++;
    }
    return 0;
}

```

Ex4023.cpp program ၏ run ရလဒ်များကို (၄-၄၂) မှတ်တမ်းတင်ပေးပါ။



(၄-၄.၂)

Passing References

→ function references reference values only function is pass reference itself calling function is argument copy if called function modify it's called function is local copy it modify it's not if (3, 4) modify it's not Ex4025.cpp program use reference pass to function modify it's not

```
Ex4025.cpp
// Listing 4.25: Passing references
#include <iostream>

struct History
{
    char name[80];
    int age;
    float height;
};

void showHistory&(History& h) // call by reference

int main()
{
    History his = {"Mark Zuckerberg", 25, 5.11};

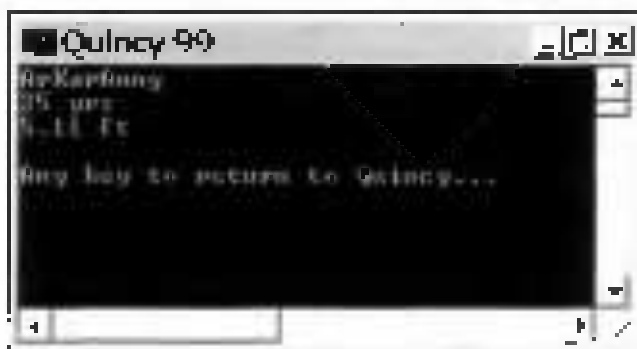
    showHis(h);
    return 0;
}

void showHistory&(History& h)
{
    cout << "Name: " << h.name << endl;
    cout << "Age: " << h.age << endl;
    cout << "Height: " << h.height << endl;
}
```

3 (3, 4)

Ex4025.cpp program အကျဉ်းချုပ်အား ကြည့်ရန်

- `show()` structure type history ကို `define` ပြုလုပ်ပြီး `show()` သို့မဟုတ် `prabotype` ကို `code` `declare` ပြုလုပ်ထားပါသည်။
- `main()` function ထဲသို့ history instance `obj`ကို `hs` ကို initialize ပြုလုပ်ပြီး `show()` function ကို `call` ပြုလုပ်ပါ။ `called` function ထဲသို့ `call` `by` `reference` ပြုလုပ်ပါသည်။ `hs` member `member`ကို `display` ပြုလုပ်ပြီး `return` ပြုလုပ်ပါသည်။
- ပုံ (၄-၅၄) ကဲ့သို့ Ex4025.cpp program ကို `run` ပြုလုပ်ပါသည်။



ပုံ (၄-၅၄)

Returning References

reference `variable` parameter `ကနေ` `function` `code` `pass` ပြုလုပ်ပါသည်။ `structure` `type` `reference` `variable` `function` `code` `return` ပြုလုပ်ပါသည်။ `example` `code` `Date` `getDate()` `function` `code` `return` ပြုလုပ်ပါသည်။ ပုံ (၄-၅၅) ကဲ့သို့ `Ex4026.cpp` program `code` `reference` `return` ပြုလုပ်ပါသည်။

Ex4026.cpp program အကျဉ်းချုပ်အား ကြည့်ရန်

- `code` `define` `structure` `type` `Date` ကို `define` ပြုလုပ်ပြီး `Date` instance `code` `birthdays`

```

Ex4D26.cpp
// Listing 4.26 Returning references
#include <ostream>

struct Date
{
    int month, day, year;
};

Date birthdays[] = { {12, 10, 1995},
                     {7, 11, 1998},
                     {3, 5, 2000},
                     {15, 7, 2001},
                     {9, 10, 2002} };

const Date& getDate(int n)
{
    return birthdays[n-1];
}

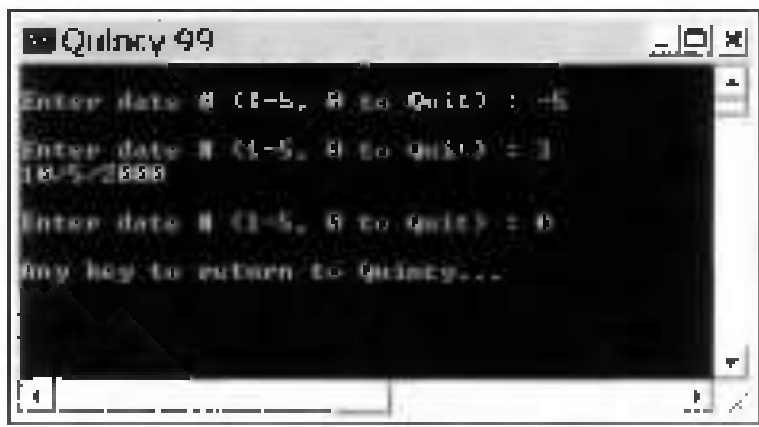
int main()
{
    int dt = 2003;
    while (dt != 0)
    {
        cout << "\nEnter date # (1-5, 0 to Quit): ";
        cin >> dt;
        if (dt > 0 && dt <= 5)
        {
            const Date& rd = getDate(dt);
            cout << rd.month << " / " << rd.day << " / "
                << rd.year << endl;
        }
    }
    return 0;
}

```

☐ (9/27)

☞ initialize `birthdays` array (Date objects) retrieve `birthdays` `getDate` function `getDate` expression

- main() function ရှေ့ကတစ်ခုခု dt ကို positive value တစ်ခုခု assign လုပ်ပေးခြင်းမရှိဘဲ while loop ထဲကိုဝင်ယိုမိပြီး dt value ကို (5) ထိုးထည့်ပြန်လျှင် if block statement ထဲဝင်ယိုမိပါသည်။ ဒါဆိုရင် dt value ကို (3) ချိတ်ကန့်ညှိသွားပါ။ if block ထဲဝင်ယိုမိပြီး const Date& rd = getDate(dt); ဆိုတဲ့ statement ကနေ local reference rd ကို returned reference getDate(dt) နဲ့ initialize လုပ်ပေးပါသည်။
- returned reference function ထဲဝင်ကတစ်ခုခု n = 3 ပြန်သွားပြီး ကော့စ် birthdays[3-1] ကို return လုပ်ပေးတဲ့အခါမှာ birthdays[2] = {10,5,2000} ကို main() ထဲက rd နဲ့ assign ပြန်လုပ်ပေးမယ်။ နောက်ပြီး rd member တွေကိုတစ်ခုချင်း display လုပ်ပေးနေပါက ဖို (၄. ၄၆) က Ex4026.cpp program ကို run ပြုထားပါ။



ဖို (၄. ၄၆)

const References

ဖို (၄. ၄၇) မှာပေါ်ပြထားတဲ့ Ex4027.cpp program မှာတော့ struct type Date ရဲ့ instance တစ်ခုဖြစ်တဲ့ dt ကို {10, 5, 2002} ဖို့ initialize လုပ်ပေးထားပါသည်။ နောက်ဖို const reference to Date တစ်ခုဖြစ်တဲ့ dr ကို dt နဲ့ assign လုပ်ပေးပြီး dr member တစ်ခုချင်းကို တွက်ပစ္စည်းမှာ display ပြန်လုပ်ပေးထားပါသည်။ Ex4027.cpp program ကို ဖို (၄. ၄၈) မှာ run ပြုထားပါသည်။

```

Ex4027.cpp

// Listing 4.27: const references
#include <iostream>

struct Date
{
    int month, day, year;
};
Date dt = {10,5,2002};

int main()
{
    const Date& dl = dt;

    cout << dl.month << '\n';
        << dl.day << '\n';
        << dl.year << endl;
    return 0;
}

```

☞ (၄. ၄၅)



☞ (၄. ၄၈)

const Reference Parameters

Ex4027.cpp program သို့ const reference parameter pass ယှဉ်အညွှန်းပြင်ပမဟုတ်အိုင် (၄. ၄၅) ဝေအဖွဲ့ဝင်အဖွဲ့အစည်းကို program run ရင်လည်းကောင်းတစ်ပြင်ပမဟုတ် (၄. ၄၈) အိုင်အဖွဲ့ဝင်

```
Ex4026.cpp

// Listing 4.28: const reference parameters

#include <ostream>
struct Date
{
    int    month, day, year;
};
const Date dt = {10,5,2002};

void    displayDate (const Date&)

int    main()
{
    const Date& dr = dt;

    displayDate (dr);
    return 0;
}

void    displayDate (const Date& x)
{
    cout << x.month << '/'
         << x.day << '/'
         << x.year << endl;
}

```

Figure 4.28

```
Quincy 99
10/5/2002
Any key to return to Quincy...

```

Figure 4.29