Chapter 9

Function I

Faculty of Engineering KMITL
King Mongkut's Institute of Technology Ladkrabang

Objective

- What is a Function?
- Types of Functions.
- Structure of Functions.
- How to Pass and Return Function statement.
- Scope and Lifetime of variables.
- How to Pass by value and Pass by reference.

What is a Function

- A function is a block of code that performs a specific task.
- Dividing a complex problem into small components improve the program understandability and usability.

Why do we have to use Functions?

- Programmers don't know all detail of the system process; Ex: Don't know the process of transferring data via network hence need to write a program to connect the network.
- Programmer don't know all procedure of computer Ex How to show the alphabet on the screen.
- Some program are complicated process and use many time Ex Science calculation and Analyze big data.

Why do we have to use Function?

Solving the problem

- The programmer who know about the detail of the system will write the programming of the process in Function format and send to others programmers.
- Programmer can call the function by knowing only how to use it and the result of it.

Ex Programmers ,who don't know how to show the alphabet on the screen, can call printf command. Programmer just know only when call this command the alphabet will appear on the screen.

Advantages of Functions

- The programmer can develop program without knowing the detail of the process function but just knowing only the result of it.
- The program will be easier to understand, maintain and debug.
- Reusable codes that can be used in other programs
- A large program can be divided into smaller modules. Hence, a large project can be divided among many programmers.

Objective

- What is a Function?
- Types of Functions.
- Structure of Function.
- How to Pass and Return Function statement.
- Scope and Lifetime of variables.
- How to Pass by value and Pass by reference.

Function types

Function can be separate into 2 types

- Standard Library function
- User Defined function

Standard Library function

- Standard Library function is already exist in the Library. Require to include directives before use.
- Directives are group of functions **Ex** stdio.h, conio.h, string.h, math.h
- Including a directive means declaring group of directives to compiler. **Ex** Using command **sin()** which in **math.h** so always have to include math.h.

Example

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
	double rad = -1.0;
	do {
		printf ( "Sine of %f is %f\n", rad, sin(rad));
		rad += 0.1;
	} while (rad <= 1.0);
	return 0;
}
```

Calling a Standard Library function

- Know the result of the program
- Know the result come from which Function
- Know directive of that Function
- Include that directive
- Call Function in the program

Example : Standard Library function

Mathematics Function

- math.h

String and Character Function

- string.h
- ctype.h

Mathematics Function

#include<math.h>	
sin(var);	
cos(var);	
tan(var);	
sqrt(var);	// Root 2
pow(var1,var2);	// $var1^{var2}$
log(var);	// $\log_e$
log10(var);	// $\log_{10}$
exp(var);	// e^{var}
fabs(var);	// absolute float

Reference..www.ce.kmitl.ac.th King Mongkut's Institute of Technology Ladkrabang

Example : Using mathematics function

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define PI 3.14
int main()
{
	float deg,rad;
	printf ("Enter Degree : ");
	scanf ("%f",&deg);
	rad = deg * PI / 180;
	printf ("sin(%.2f) = %.3f\n",deg,sin(rad));
	printf ("cos(%.2f) = %.3f\n",deg,cos(rad));
	printf ("tan(%.2f) = %.3f\n",deg,tan(rad));
	return 0;
}
```

String Function

#include<string.h>

strcpy(str1, str2);
strcat(dest1, src2);
strcmp(dest1, src2);
strcmpi(str1, str2);
strlen(str);

#include<ctype.h>
tolower(ch); toupper(ch);

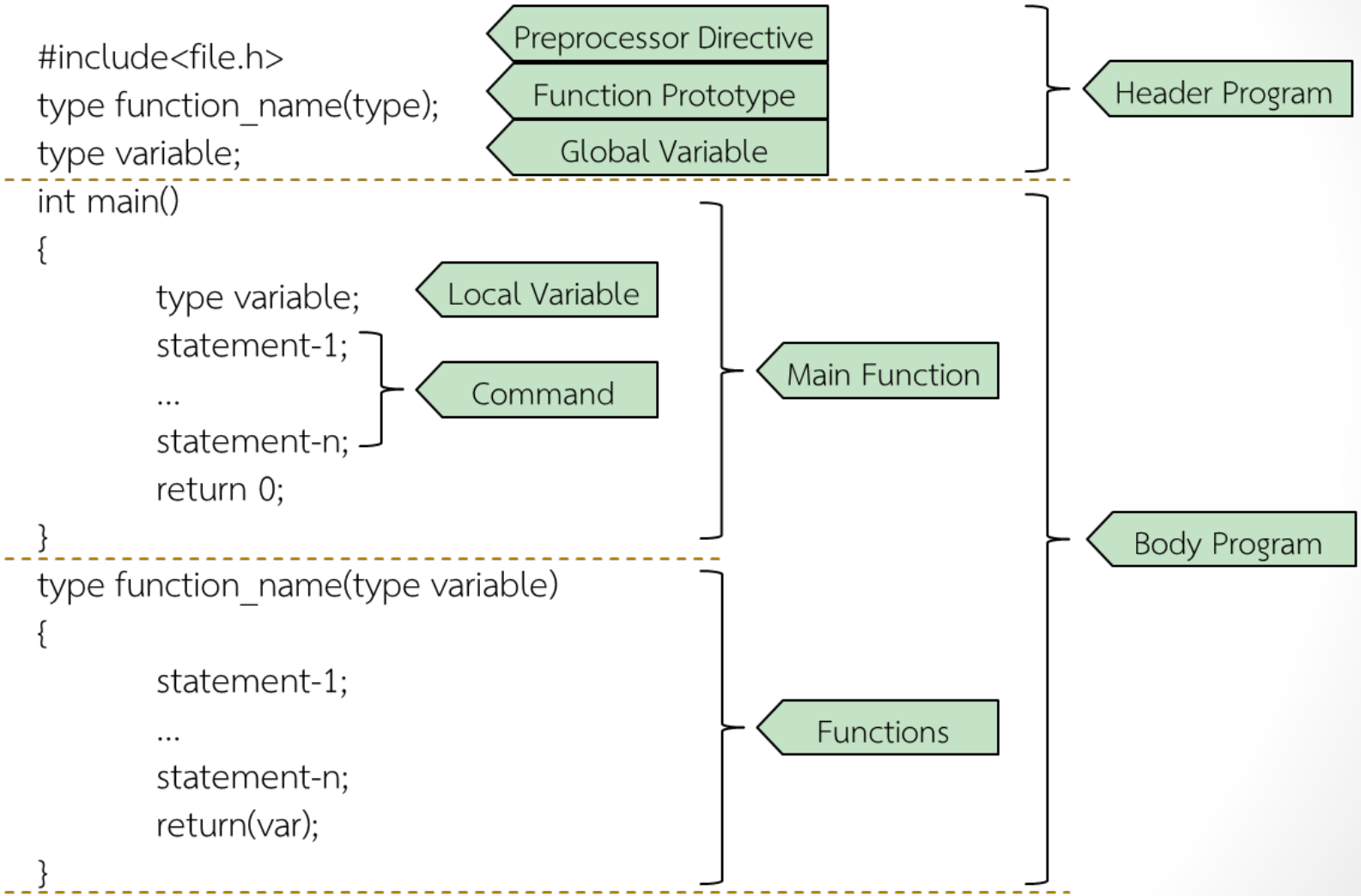
User-defined Functions

- Standard Library Function are only basic function.
- If programmer wants to use special function, programmers have to create it by themselves.

Basic Requirements of User-defined Functions

- Declaration function prototype at the beginning of the program before call each Function (Inform the compiler that Function is not syntax error).
- Write the Function following the structure that declares at Function Prototype.

C Language Structure

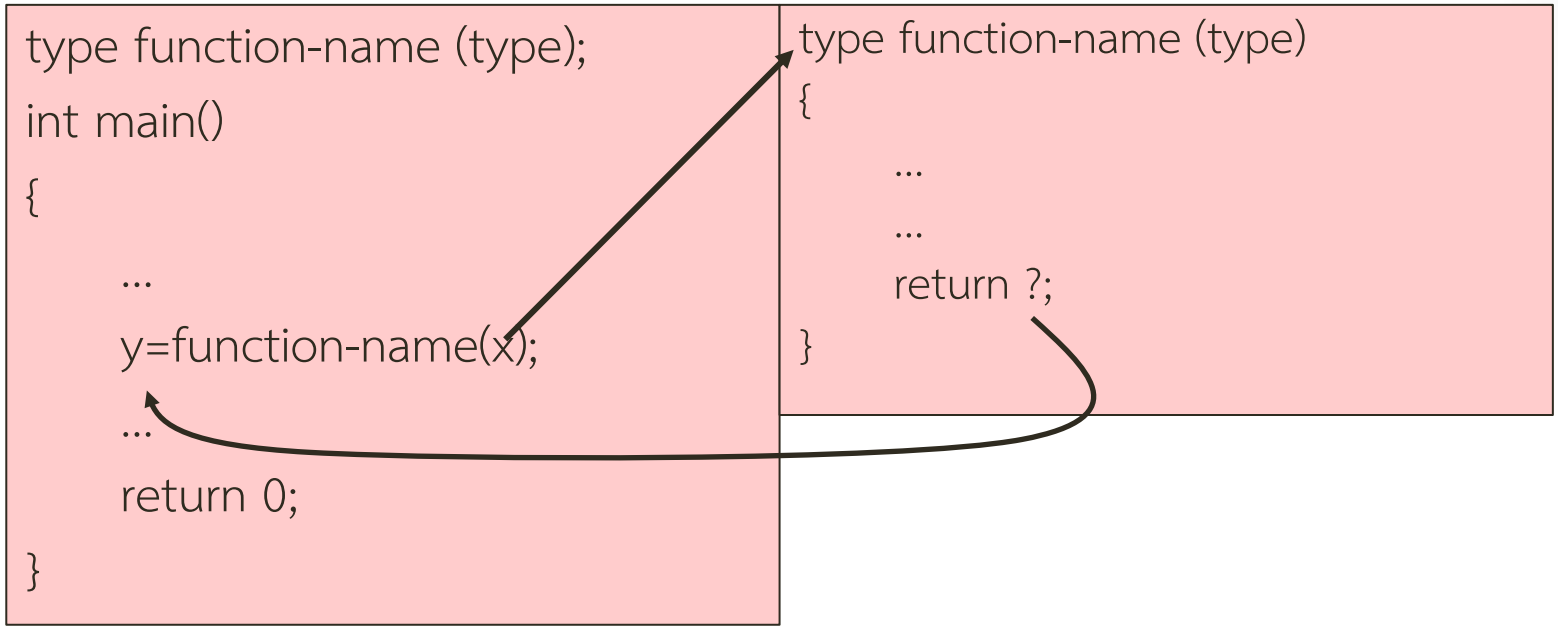


Reference..www.ce.kmitl.ac.th King Mongkut's Institute of Technology Ladkrabang

Objective

- What is a Function?
- Types of Functions.
- Structure of Function.
- How to Pass and Return Function statement.
- Scope and Lifetime of variables.
- How to Pass by value and Pass by reference.

User-defined Functions



Function Prototypes

Declaration function after main()

```
type function_name (type-1, type-2, ..., type-n);
```

type is type of the function that define the return value type.

function_name is name of function

type-n is type of data that will send to function (Use for being the input of function)

User-defined Functions

Functions can be created with 2 types.

- Create function before main function.
  - Main function can use created function.
- Create function after main function.
  - Have to declare Function Prototype before main function which main function will know the created function.

Create function before main function

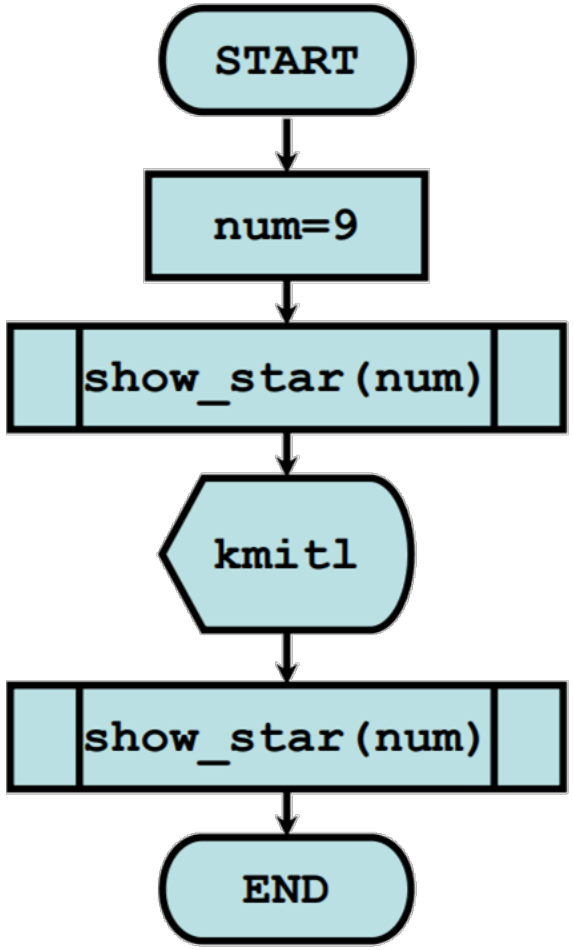
```
#include<file.h>
type variable
type function_name(type variable)
{
        statement-1;
        ...
        statement-n;
        return(var);
}
int main()
{
        type variable;
        statement-1;
        ...
        statement-n;
        return 0;
}
```

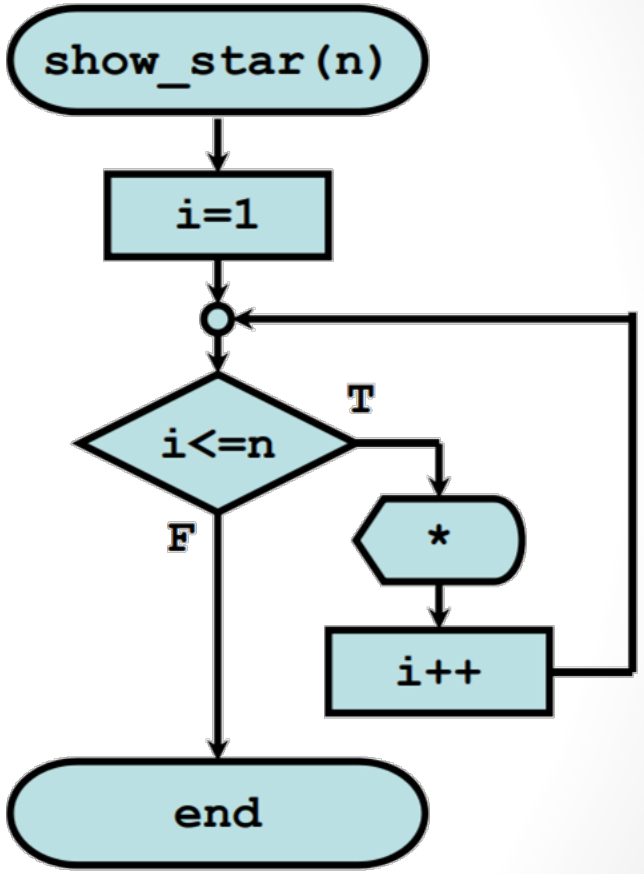
Create function after main function

```
#include<file.h>
type function_name(type variable);
type variable
int main()
{
        type variable;
        statement-1;
        …
        statement-n;
        return 0;
}
type function_name(type variable)
{
        statement-1;
        …
        statement-n;
        return(var);
}
```

Example : Flowchart and Result

```
*********
* kmitl *
*********
```





Example

```
#include<stdio.h>
#include<conio.h>
void show_star (int n)
{
	int i;
	for (i=1;i<=n;i++)
	putchar('*');
}
int main()
{
	int num=9;
	show_star(num);
	printf ("\n* kmitl *\n");
	show_star(num);
	return 0;
}
```

```
#include<stdio.h>
#include<conio.h>
void show_star (int);
int main()
{
	int num=9;
	show_star(num);
	printf ("\n* kmitl *\n");
	show_star(num);
	return 0;
}
void show_star (int n)
{
	int i;
	for (i=1;i<=n;i++)
	putchar('*');
}
```

Objective

- What is a Function?
- Types of Functions.
- Structure of Function.
- How to Pass and Return Function statement.
- Scope and Lifetime of variables.
- How to Pass by value and Pass by reference.

Type of User-defined Functions

- Functions with no arguments and no return value.
- Functions with arguments and no return value.
- Functions with arguments and a return value.
- Functions with no arguments and a return value.

*Function type can separated by function prototype.

Type of User-defined Functions

- Functions with no arguments and no return value.

```
#include<stdio.h>
#include<conio.h>
void function_name(void);
int main()
{
        ...
        function_name();
        ...
        return 0;
}
```

```
void function_name()
{
        statement-1;
        statement-2;
        ...
        statement-n;
}
```

Example : Functions with no arguments and no return value.

```
#include<stdio.h>
void PrintHello(void);
int main()
{
        PrintHello();
        printf("Hello,in function main.\n");
        PrintHello();
        return 0;
}
void    PrintHello(void)
{
    printf("Hello,in function PrintHello ..\n\n");
}
```

Type of User-defined Functions

- Functions with arguments and no return value.

```
#include<stdio.h>
#include<conio.h>
void func_name(type);
int main()
{
        ...
        func_name(varX);
        ...
        return 0;
}
```

```
void func_name(type varY)
{
        statement-1;
        statement-2;
        ...
        statement-n;
}
```

Example : Functions with arguments and no return value.

```
#include<stdio.h>
void PrintHello( int );
int main()
{           int n;
            printf("input number of hello : ");
            scanf("%d",&n);
            PrintHello( n );
            return 0;
}
void PrintHello( int i )
{     int count;
      for ( count=1; count<=i ; count++ )
            printf("%d HELLO\n",count);
}
```

Type of User-defined Functions

- Functions with arguments and return value.

```
#include<stdio.h>
#include<conio.h>
type func_name(type);
int main()
{
        ...
        var = func_name(varX);
        ...
        return 0;
}
```

```
type func_name(type varY)
{
        statement-1;
        statement-2;
        ...
        statement-n;
        return(varZ);
}
```

Example : Functions with arguments and return value.

```
#include<stdio.h>
float CircleArea( int );
int main()
{       int radius;
        printf("input radius : "); scanf("%d",&radius);
        printf(" Circle area = %f\n",CircleArea(radius));
        return 0;
}
float CircleArea( int rad )
{       float answer=0;
        answer = 22.0/7.0*rad*rad ;
        return answer;
}
```

Type of User-defined Functions

- Functions with no arguments and return value.

```
#include<stdio.h>
type func_name(void);
int main()
{
...
var = func_name();
...
return 0;
}
```

```
type func_name()
{
statement-1;
statement-2;
...
statement-n;
return(varZ);
}
```

Example : Functions with no arguments and return value.

```
#include<stdio.h>
int RoundInput( void );
int main()
{       int i,round;
        round = RoundInput();
        for( i=1; i<=round;i++) printf("hello #%d\n",i);
        return 0;
}
int RoundInput(void)
{       int answer;
        printf( "Please input number of hello : ");
        scanf( "%d", &answer );
        return answer;
}
```

Example

From the example program please describe.

- What is the name of the Function?
- Is there argument value?, If yes how many parameter?
- What is the variable type of argument value?
- There is return value or not, If yes how many parameters?
- If the programmer want to remove line /*Function Prototype*/, How to adjust the program.
- How does the program work?

Example

```
#include<stdio.h>
#include<math.h>
#define PI 3.14
float deg_rad(float); /*Function Prototype*/
int main()
{
        float deg,rad;
        printf ("Enter Degree : ");
        scanf ("%f",&deg);
        rad = deg_rad(deg); //printf ("%f->%f\n",deg,rad);
        printf ("sin(%.2f) = %.3f\n",deg,sin(rad));
        printf ("cos(%.2f) = %.3f\n",deg,cos(rad));
        printf ("tan(%.2f) = %.3f\n",deg,tan(rad));
        return 0;
}
float deg_rad(float num)
{
        float ans;
        ans = num * PI / 180;
        return(ans);
}
```

Example

From the example program please describe.

- What is the name of the Function?
- Is there argument value?, If yes how many parameter?
- What is the variable type of argument value?
- There is return value or not, If yes how many parameters?
- If the programmer want to remove line /*Function Prototype*/, How to adjust the program.
- How does the program work?

Reference..www.ce.kmitl.ac.th King Mongkut's Institute of Technology

Exercise

1. Write a program to calculate triangle area by get 3 sides (a, b, c) of it from keyboard.

Formula

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{a + b + c}{2}$$

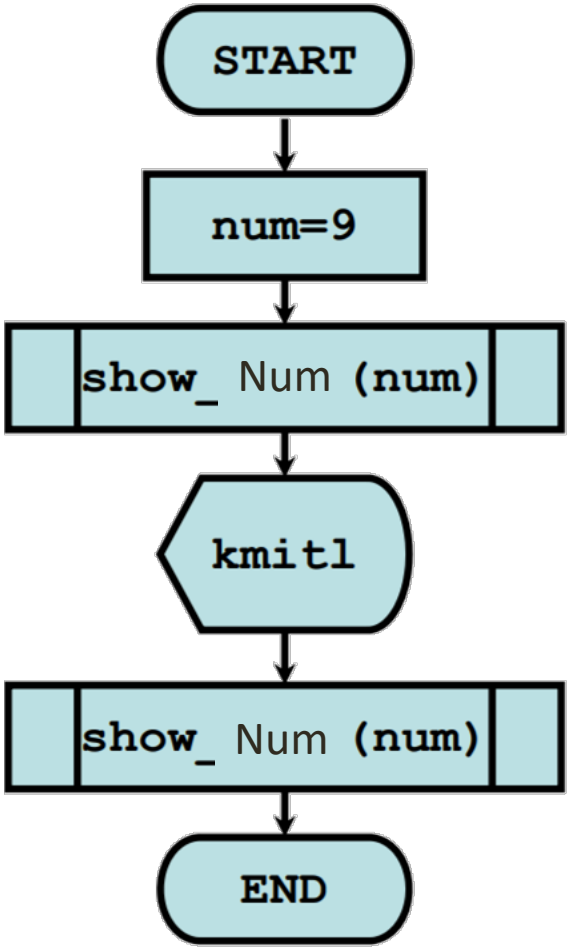
2. Write a program to calculate exchange US dollar and Thai baht. Get dollar value from keyboard and show the result in Thai baht. Using function D2B (int) for calculate dollar to Thai baht Define 1 dollar = 31.25 baht.

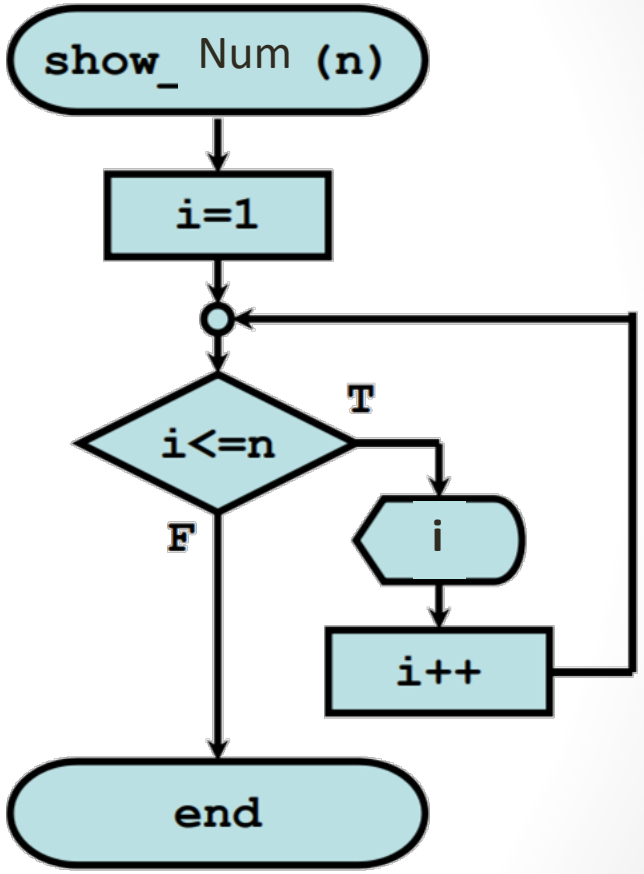
3. Below is the part of a program use for check an input(Key board input)number is a prime number or not.

Please complete the program.

Example : Flowchart and Result

```
123456789
* KMITL *
123456789
```





Exercise

1. Write a program to calculate triangle area by get 3 sides (a, b, c) of it from keyboard.

Formula

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{a + b + c}{2}$$

Reference..www.ce.kmitl.ac.th King Mongkut's Institute of Technology Ladkrabang

Exercise

```
#include<stdio.h>
int main()
{
        int N;
        printf("Input N: ");
        scanf("%d",&N);
        if ( IsPrime(N) )
                printf("%d is Prime number ",N );
        else
                printf("%d is not Prime number ",N );
        return 0;
}
```

Objective

- What is a Function?
- Types of Functions.
- Structure of Function.
- How to Pass and Return Function statement.
- Scope and Lifetime of variables.
- How to Pass by value and Pass by reference.

Variable and scope of Functions

- **Global variable** is a variable that all functions can use. Declare variables after Preprocessor directive.
- **Local variable** is a variable that can use only declared function. Declare variables within that function.

```
#include<stdio.h>
int num1; // num1 is global variable
void test(void);        /*Function Prototype*/
int main()
{
        num1 = 19;      // no num1 declaration
        printf ("line1 (main) : num1 = %d\n",num1);
        test();
        printf ("line2 (main) : num1 = %d\n",num1);
        return 0;
}
void test()
{
        num1 = 26;      // no num1 declaration
        printf ("line1 (test) : num1 = %d\n",num1);
}
```

```
line1 (main) : num1 = 19
line1 (test)  : num1 = 26
line2 (main) : num1 = 26
```

```
#include<stdio.h>
void test(void);        /*Function Prototype*/
int main()
{
        int num1;       // local num1 in main()
        num1 = 19;
        printf ("line1 (main) : num1 = %d\n",num1);
        test();
        printf ("line2 (main) : num1 = %d\n",num1);
        return 0;
}
void test()
{
        int num1;       // local num1 in test()
        num1 = 26;
        printf ("line1 (test) : num1 = %d\n",num1);
}
```

```
line1 (main) : num1 = 19
line1 (test)  : num1 = 26
line2 (main) : num1 = 19
```