Steps to solve logical and numerical problems

PRESENTED BY: SATYA PRAKASH PATEL
EMAIL: SATYAPATEL.IND@GMAIL.COM

Procedure :Steps Involved in Problem Solving

In order to solve a problem by the computer, one has to pass though certain stages or steps. They are

1. Understanding the problem
2. Analyzing the problem
3. Developing the solution
4. Coding and implementation.

Algorithms

A set of sequential steps usually written in Ordinary Language to solve a given problem is called **Algorithm**.

Or **“a complete, unambiguous, finite number of logical steps for solving a specific problem “**

Steps involved in algorithm development

Step1. Identification of input: The input is to be indentified first for any specified problem.

Step2: Identification of output: From an algorithm, at least one quantity is produced, called for any specified problem.

Step3 : Identification the processing operations: All the calculations to be performed in order to lead to output from the input are to be identified in an orderly manner.

Step4 : Processing Definiteness: The instructions composing the algorithm must be clear and there should not be any ambiguity in them.

Step5 : Processing Finiteness: If we go through the algorithm, then for all cases, the algorithm should terminate after a finite number of steps.

Step6 : Possessing Effectiveness : The instructions in the algorithm must be sufficiently basic and in practice they can be carries out easily

An algorithm must possess the following properties

1. Finiteness: An algorithm must terminate in a finite number of steps

2. Definiteness: Each step of the algorithm must be precisely and unambiguously stated

3. Effectiveness: Each step must be effective, (in the sense that it should be primitive easily convert able into program statement) can be performed exactly in a finite amount of time.

4. Generality: The algorithm must be complete in itself so that it can be used to solve problems of a specific type for any input data.

5. Input/output: Each algorithm must take zero, one or more quantities as input data produce one or more output values. An algorithm can be written in English like sentences or in any standard representation sometimes, algorithm written in English like languages are called Pseudo Code

Example

find the average of three numbers
Step 1 Read the numbers a, b, c
Step 2 Compute the sum of a, b and c
Step 3 Divide the sum by 3
Step 4 Store the result in variable d
Step 5 Print the value of d
Step 6 End of the program

Flowcharts: Already discussed

Pseudo code

The Pseudo code is neither an algorithm nor a program. It is an abstract form of a program. It consists of English like statements which perform the specific operations.

Advantages: * Easy to read, * Easy to understand, * Easy to modify.

Example: Write a pseudo code to perform the basic arithmetic operations.

Read n1, n2

Sum = n1 + n2

Diff = n1 – n2

Mult = n1 * n2

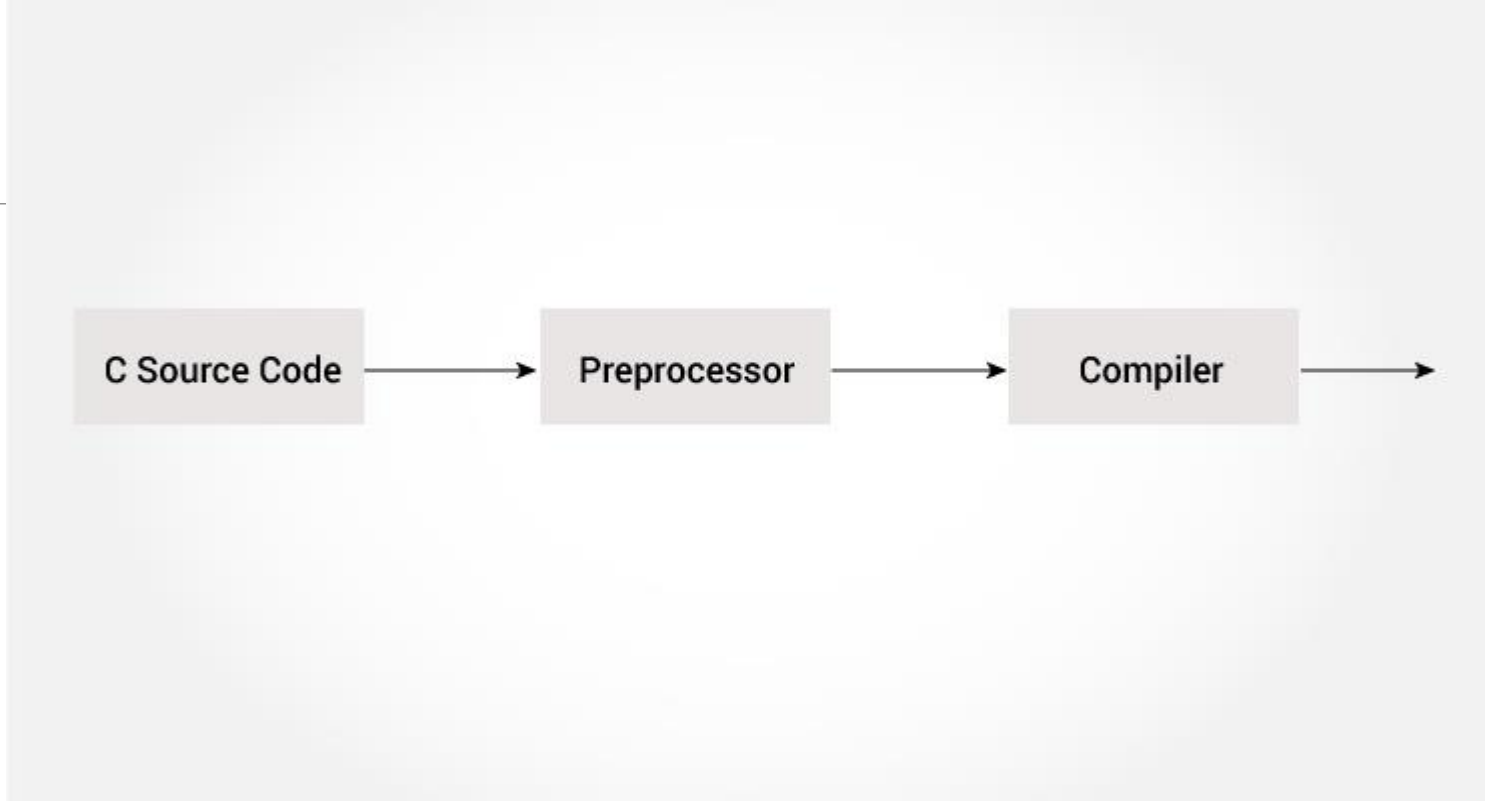
Quot = n1/n2

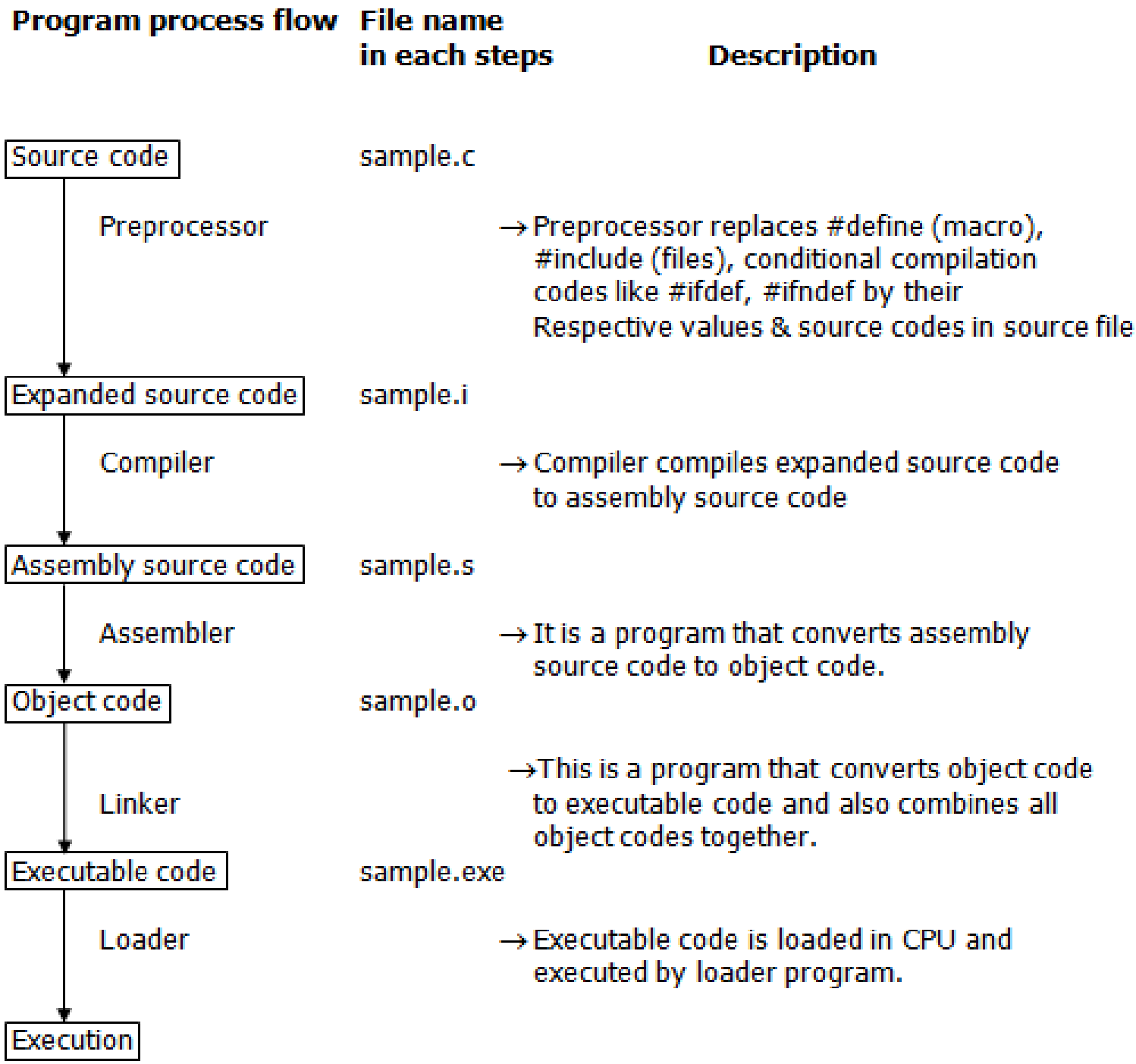
Print sum, diff, mult, quot

End.

C – Preprocessor directives

1. Before a C program is compiled in a compiler, source code is processed by a program called pre-processor. This process is called pre-processing
2. Commands used in pre-processor are called pre-processor directives and they begin with “#” symbol.


C Source Code
Preprocessor
Compiler


Program process flow
File name in each steps
Description
Source code
sample.c
Preprocessor
→ Preprocessor replaces #define (macro), #include (files), conditional compilation codes like #ifdef, #ifndef by their Respective values & source codes in source file
Expanded source code
sample.i
Compiler
→ Compiler compiles expanded source code to assembly source code
Assembly source code
sample.s
Assembler
→ It is a program that converts assembly source code to object code.
Object code
sample.o
Linker
→ This is a program that converts object code to executable code and also combines all object codes together.
Executable code
sample.exe
Loader
→ Executable code is loaded in CPU and executed by loader program.
Execution

Predefined Macros

Predefined macro	Value
__DATE__	String containing the current date
__FILE__	String containing the file name
__LINE__	Integer representing the current line number
__STDC__	If follows ANSI standard C, then value is a nonzero integer
__TIME__	String containing the current date.

Get current time using __TIME__

```
#include <stdio.h>

int main()

{

   printf("Current time: %s",__TIME__);   //calculate the current time

}
```

Preprocessor	Syntax/Description
Macro	**Syntax:** #define This macro defines constant value and can be any of the basic data types.
Header file inclusion	**Syntax:** #include <file_name> The source code of the file “file_name” is included in the main program at the specified place.
Conditional compilation	**Syntax:** #ifdef, #endif, #if, #else, #ifndef Set of commands are included or excluded in source program before compilation with respect to the condition.
Other directives	**Syntax:** #undef, #pragma #undef is used to undefine a defined macro variable. #Pragma is used to call a function before and after main function in a C program.

Run time error in C

A runtime error is a program error that occurs while the program is running.

Class Problem 1

```
#include<stdio.h>
#define SWAP(a, b)( a=a+b, b=a-b, a=a-b)
int main()
{    int x=10, y=20;
     SWAP(x, y);
     printf("%d %d\n", x, y);
     return 0;
}
```

```
#include<stdio.h>
#define int char
main()
{
        int i=5;
        printf ("sizeof (i) =%d", sizeof (i));
}
```

Question 1

What will be the output of following program ?

Question 2

In which stage the following code #include<stdio.h> gets replaced by the contents of the file stdio.h

A. During editing
B. During linking
C. During execution
D. During preprocessing

Question 3

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include<stdio.h>
int main()
{
        int i, j;
        for(i = 1, j = 1;i<=3,j<=3;i++,j++)
        printf("%d %d ",i, j);
        return 0;
}
```

OPTIONS

A. Compilation Error

B. 1 2 3 1 2 3

C. 1 1 2 2 3 3

D. None of the above

Question 4

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include<stdio.h>
#define loop for(;;)
int main()
{
        printf(“F.R.I.E.N.D.S");
        loop;
        return 0;
}
```

OPTIONS

A. Compilation error

B. F.R.I.E.N.D.S

C. Program never ends

D. None of the above

Question 5

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include <stdio.h>

int main()

{int i;
     if (printf("0"))

         Printf(" Coooooooooooool");

    else

         i = 5;

    printf("%d", i);

    return 0;

}
```

OPTIONS

(A) Runtime Error
(B) 0 Coooooooool
(C) Coooooooool

(D) None

Queries and Feedback