C Preprocessor

The preprocessor, as its name implies, is a program that processes the source code before it passes through the compiler. It operates under the control of what is known as preprocessor directives.The preprocessor acts on the source program according to the instructions specified to the source program and the output produced by the preprocessor is then submitted to the C compiler. The instructions specified in the source program to the preprocessor are called preprocessor directives.

General rules for preprocessor directives

1. All preprocessor directives should start with the symbol #.
2. only one directive can appear in a line
3. the directives are not terminated by semicolon
4. the preprocessor directives can be placed anywhere in the source program

Classification

Based on the purpose, the preprocessor directives are classified into 4 types

1. file inclusion directives
2. macros definition directives
3. conditional compilation directives
4. miscellaneous directive

File inclusion:

examples

#include <stdio.h>

Or

#include “stdio.h”

The preprocessor replaces this line by the contents of the file name.

Macro definition

1. #define PI 3.14
2. #define BUFFER_SIZE 10
3. #define OR ||
4. #define PF printf

Conditional compilation directives

C preprocessor offers a feature known as conditional compilation, which can be used to switch on or off a particular line or group of lines in a program. This is achieved by inserting the preprocessor commands #ifdef, #endif.

The general form is

```
#ifdef macroname
statement1;
statement2;
statement3;
#endif
```

if the macro has been #define, the block code will be processed as usual, otherwise not.

Example

```
main()
{
#ifdef OKAY
statement 1;
statement 2;
#endif
statement3;
}
```

here, statements 1 and 2 would get compiled only if the macro OKAY has been defined and the definition of the macro has been purposely omitted.

#ifndef (if not defined) it works exactly opposite of #ifdef

Program: macro with arguments

```
#include<stdio.h>
#define AREA(r) 3.14*(r)*(r)
int main()
{

int r;
float area;
printf("Enter the radius");
//scanf("%d",&r);
//area=AREA(r);
area=AREA(1+1); //3.14 * 1+1 * 1+1
printf("Area of circle is %f",area);
return 0;
}
```

Program for conditional compilation

```
// #define MEMBER
// #define PREMIUM_MEMBER

int main()
{
    float amount, total;
    printf("Enter amount:\n");
    scanf("%f", &amount);

    #ifdef MEMBER
        total = amount * 0.8; // 20% discount for MEMBER
        printf("Amount to be paid = %f", total);
    #elif defined(PREMIUM_MEMBER)
        total = amount * 0.7; // 30% discount for PREMIUM_MEMBER
        printf("Amount to be paid = %f", total);
    #else
        printf("Amount to be paid = %f", amount); // No discount for non-members
    #endif

    return 0;
}
```

Code Explanation:

1. The #ifdef MEMBER condition is evaluated first. If neither MEMBER nor PREMIUM_MEMBER is defined, the #else block executes, providing no discount.
2. #elif defined(PREMIUM_MEMBER) checks for another preprocessor directive PREMIUM_MEMBER to apply a 30% discount.

3. Uncomment or define #define MEMBER or #define PREMIUM_MEMBER as needed to test different conditions.