C program passes through 3 phases.

1. Pre- processing,
2. Compiling,
3. Linking
   - ⇨ exe generate.

Preprocessor is a program which pre processes your program before it compiles.

We use pre-processor directives or pre-processor command to pre process.

pre-processor directives

source code are being examine before compilation if it has any pre-processor directies.

Preprocessor Command
They starts with # command.
They are not program lines and always put in first column in source cod.
No semicolon (;) needed to put last for preprocessor command.

Eg.
#include<stdio.h>
[include stdio.h file before compiling. File inclusion command]

#define used to define macro.

#undef used to undefined macro.
#if
#else
#endif

There are 3 categories pre processor directives :

1. **Macro substitution directives (used for macro)**
2. **File inclusion directives (used for including file)**
3. **Compiler control directives (for controlling condition during compilation .. eg. Skiping the f line.)**

Macro substitution directives (used for macro)

We are defining substitute value for identifier.
Syntax :
#define identifier value

Common form where to use macro:

- Simple Macro substitution :
- Argumented Macro substitution
- Nested Macro substituion

Simple Macro substitution

```
#define VALUE 10
d= VALUE – 5; // (10-5)

#define AREA 50*2

#define SIZE sizeof(int)*4


#define A 50-10
#define B 10-2
C = A * B;    //C= 50-10 * 10-2);  // according operator precedent  multiplication will be done before subtraction.
```

Problem

Right way :

```
#define A (50-10)
#define B (10-2)
C = A * B;    // 40 * 8
```

[make sure you always use bracates () if you use expression in macro.

```
#include<stdio.h>
#define LIKES 10000
#define MESSAGE “Hi”
int main()
{
int   n = 2000;
      if (LIKES>n)
      {
      printf(MESSAGE);
      }
return 0;
}
```

<u>Compiler Control Directives</u>

Condtional compilation : if same programme for two different machine than you can skip few lines during compilation.

```
#ifdef // uses
#else
#endif
#if (
```

****

Ex 1.

```
#define X 100 // stored in macro.c
#ifndef X
#define X 200 // as X is already defined in macro.c so this line will be skipped.
#endif
```

************Ex 2**************************

```
#define y 100 // stored in macro.c
#ifndef X
#define X 200 // this line will be executed in this case as X is not defined earlier.
#endif
```

<u>**File inclusion directives (used for including file)**</u>

<u>**Compiler control directives (for controlling condition during compilation .. eg. Skiping the f line.)**</u>