PreProcessor directives

↳ #include → #include <stdio.h>
↳ text RelPlacement
for Conent about stdio.h

↳ include standard lib → < >
↳ ex <stdio.h> | <stdlib.h>
<stdint.h>

↳ include user lib → " "
↳ ex → Array.h | Math.h
↳ include way → Absoulte Path

#include "C:/Heshan/E.S/NTI/Day5/std.h"

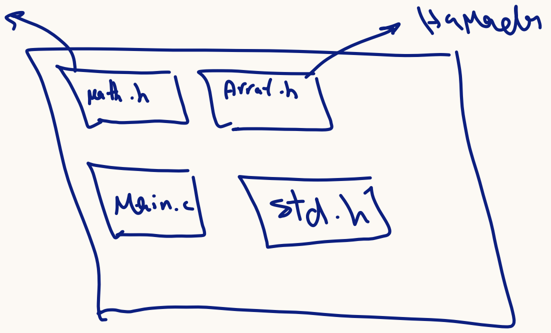
File.c → PreProcessor → File.i

* text Replacement for preprocessor directive (#)

* gcc -E File.c -o File.i

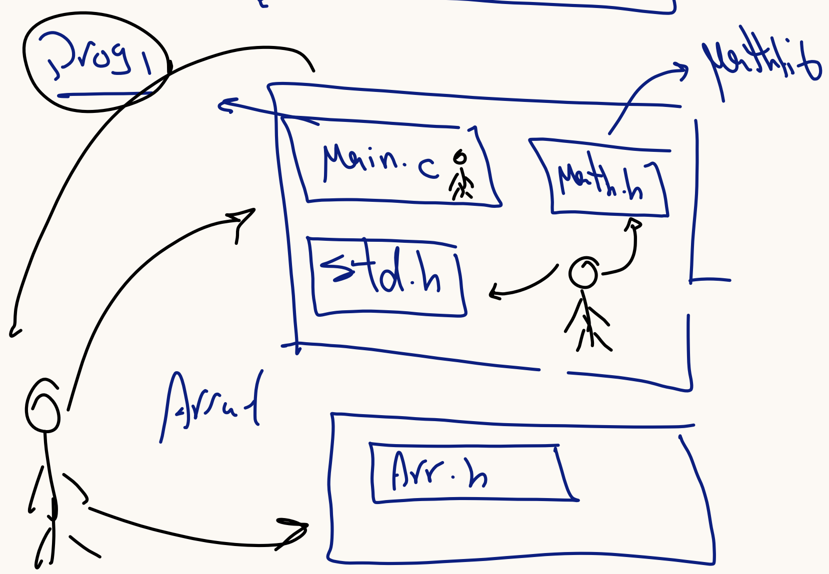
#include | #define
#undef | #if #elif
#else #endif #warning
#error #ifndef
#ifdef .

Mathlib → Relative Path



Main.c

#include "std.h"
#include "Mathlib/Math.h"
#include "Hamada/Array.h"



Main.c

#include "../std.h"
#include "../Mathlib/Math.h"
#include "../../Array/Array.h"

① Develop time
② Compilation
③ Runtime (exe)

#define → Macro

object like Macro

No have space memory

```
#define size 20
int arr [size]
for(i=0; i<size; i++
{
}
for(i=0; i<size; i++
{
}
```

Pre Processor

```
int arr [20]
for(i=0; i<20;
```

int x → intsize

```
int arr[5];
for(i=0; i<5; i++
{
}
for(i=0; i<5; i++)
```

→ Magic Number

---

ex

Code.C

```
#define size 5
int main ( )
{ int arr [size];
  for(i=0; i<size; i++)
  {
  }
  for(i=0; i<size-1; i++)
  {
  }
}
```

Pre Processor → Compiler

```
int main
{
 int arr [5]
 for(i=0; i<5; i++)
 {
 }
 for(i=0; i<4; i++)
 {
 }
}
```

notes object like macro

① effect line line after write #define

```
int arr[size]=0;
#define size 5
for(i=0; i<size; i++)
```

② we can't write object have the same name for var

```
#define size 5
int main()
{
    int size = 20;
}
```

Pre Processor

```
int 5 = 20;
```

Compiler error

Function like macro

```
#define Add(x, y) x+y
```

No space in memory

```
int main
{
    int sum = Add(5, 7);
}
```

Pre Processor →

```
int sum = 5+7;
```

#define PrintCV( ) Printf("Hesken\n")

```
int Main( )
{
    PrintCV( );
    PrintCV( );
    PrintCV( );
    PrintCV( );
}
```

PreProcessor

```
Printf
Printf
Printf
Printf
```

10 Byte

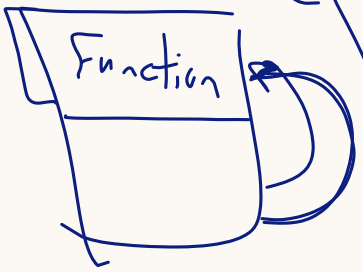
Normal Function

```
void Function ( )
{Printf ( "\n" );
}
int Main( )
{
    Function( );
    Function( );
    Function( );
    Function( );
}
```

Function like Macro

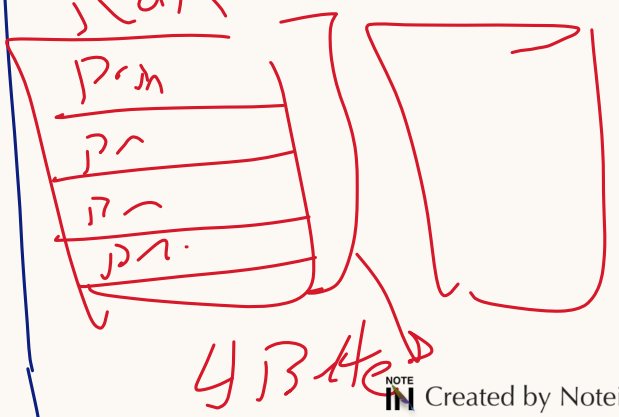
```
#define Function( ) Printf(
int Main( )
{
    Function( ) Print
    Function( ) Print
    Function( ) Print
    Function( ) Print
}
```

RAM 1B-Byte



Ram

RAM



4 Byte

Normal Function Save Memory
take time execution

Macro Function
take large memory
From memory
Fast execution

#define u8
object
char

char → PreProcessor
Bad SW

typedef → Rename Data type

typedef size 20; → Compiler

```
#define Print()   printf("Hesham\n");\
                  printf("Ahmed\n");\
```

#if #elif #else #endif

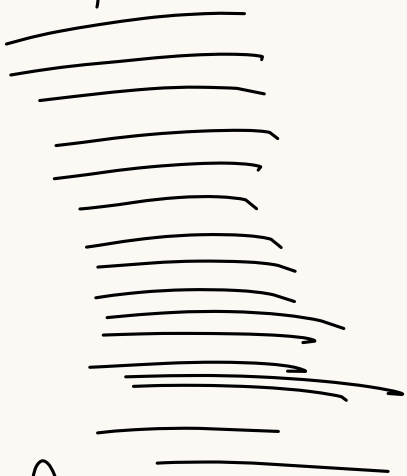
Code → Carversion → 1, 2, 3, 4

```
#define Carversion 1

int Main(){
#if Carversion == 1
  Printf("version1");
#elif Carversion == 2
  Printf("version2");
#elif Carversion == 3
  Printf("version3");
#elif Carversion == 4
  Printf("version4");
#else
  Printf("No version");
#endif
```

PreProcessor → Filter

```
int Main()
{
  Printf("version1");
}
```

```
#define Motor type 1

#if Motor type == 1
```



```
#elif Motor type == 2
```