[C test]

1. Explain "#error"

#error指令

TAG:預編譯和宏定義,ANSI C,#error

TEXT:

**語法格式如下：**

#error token-sequence

其主要的作用是在編譯的時候輸出編譯錯誤信息token-sequence，從方便程序員檢查程序中出現的錯誤。例如下面的程序

#include "stdio.h"

int main(int argc, char\* argv[])

{

#define CONST\_NAME1 "CONST\_NAME1"

printf("%s\n",CONST\_NAME1);

#undef CONST\_NAME1

#ifndef CONST\_NAME1

#error No defined Constant Symbol CONST\_NAME1

#endif

{

#define CONST\_NAME2 "CONST\_NAME2"

printf("%s\n",CONST\_NAME2);

}

printf("%s\n",CONST\_NAME2);

return 0;

}

在編譯的時候輸出如編譯信息

fatal error C1189: #error : No defined Constant Symbol CONST\_NAME1

2. Explain "struct" and "union"

聯合為早期 C 語言為了解決記憶體不足而有的資料型態，其定義與宣告與結構類似，不同之處為聯合的所有成員佔用相同的記憶體空間，因此儲存到聯合的值會以該成員具有儲存範圍最大的資料型態為優先，也就是佔用到最多位元組數的資料型態

3. Explain "volatile". Can we use "const" and "volatile" in the same variable? Can we use "volatile" in a pointer?

 Group1:     
     int a = 0;      
     const int\* b = &a;------------  [1]                  
     int const \*b = &a;------------  [2]                       
     const int\* const b = &a;---- [4]     
        [1][2]相同 都是\*b不可改但b可改

[4]\*b和b都不可改因為兩個const一個在左一個在右  
    Group2:    
     const char \*p = "const";--------------[1]   
     char const \*p = "const";--------------[2]     
     char\* const p = "const";--------------[3]     
     const char \* const p = "const";----[4]        
      [3] p的指向不可改(p=&a 錯 \*p=4 ok)  
     Group3:  
      int a=0;  
        const int &b = a;---------------[1]  
     int const &b = a;---------------[2]    
     int & const b = a;--------------[3]  //---> 修饰引用时 ,const 被忽略   
     const int & const b = a;-----[4]

4.

unsigned long v1 = 0x 00001111;

unsigned long v2 = 0x 00001202;

unsigned long v;

v = v1&(~v2);

v = v | v2;

ask: the value of v?

5.

int a[5] ={1,2,3,4,5};

int \*p = (int \*)(&a+1);

ask: the value of \*(a+1), (\*p-1)?

6. write a code

a) set the specific bit

b) clear the specific bit

c) inverse the specific bit (0->1; 1->0)

7. Re-write void(\*(\*papf)[3])(char \*);

typedef\_\_\_\_\_\_\_\_\_\_;

pf(\*papf)[3];

typedef void\* pf

8. write a code that check the input is a multiple of 3 or not without using division or mod

9. Explain lvalue and rvalue.

※ 引述《yoco315 (眠月)》之銘言：

: ※ 引述《hanck ( )》之銘言：

: : void main(int n)

: : {

: : if n==33 execute func1;

: : if n==67 execute func2;

: : if n==324 execute func3;

: : if n==231 execute func4;

: : if n==687 execute func5;

: : }

: void 麵 () {

: void (\*f[8])() = { 0, func1, 0, func2, func3, func4, 0, func5 } ;

: f[n%8]() ;

: }