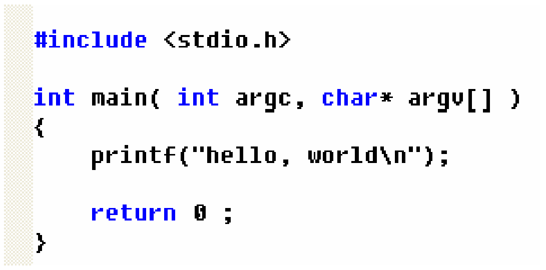
2008 summer course   C-language   Homework 1

Exercise 1: follow the instructions as we say in the course, write a “HelloWorld” program with the aid of visual studio IDE and compile, execute it.



```
#include <stdio.h>

int main( int argc, char* argv[] )
{
    printf("hello, world\n");

    return 0 ;
}
```

Figure 1: source code of “HelloWorld” program.

Exercise 2: in page 8 of textbook, the author says that ***\n*** represents only a single character (字元), in fact, we call single character **‘\’** as escape character (脫逸字元). ***\n*** means that escape character \ escapes original meaning of character n, into second meaning of n, line feed (換行).

Table 1: complete set of escape sequence in page 38 of textbook

Escape sequence	Meaning	Escape sequence	meaning
\a	Alert (bell) character	\\	Backslash
\b	Backspace	\?	Question mark
\f	Formfeed	\’	Single quot
\n	Newline	\”	Double quot
\r	Carriage return	\ooo	Octal number
\t	Horizontal tab	\xhh	Hexadecimal number
\v	Vertical tab		

In your “HelloWorld” program, try the following modification

(1) Try printf("hello, world");

(2) Try printf("hello \t world\n");

(3) Try printf("hello \\ world");

for each case, you need to recompile your program and execute it.

Exercise 3: in page 14 of textbook, the author introduces a new keyword “symbolic constants” in section 1.4, the format is

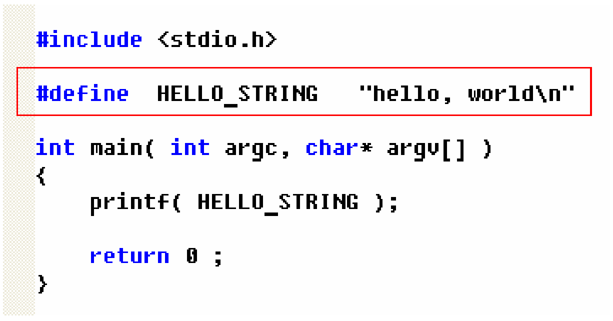
    #define   name   replacement text

modify your “HelloWorld” program like Figure 2 and check its execution result, does the result is the same as that in **Exercise 1**?

In Figure 2, we call HELLO_STRING as macro (巨集), it represents string “hello, world\n”, or you can say HELLO_STRING = “hello, world\n”.

In fact before compiler compiles the “HelloWorld” program, it will call preprocessor (前處理器)

to do macro substitution. Later on we will show this.

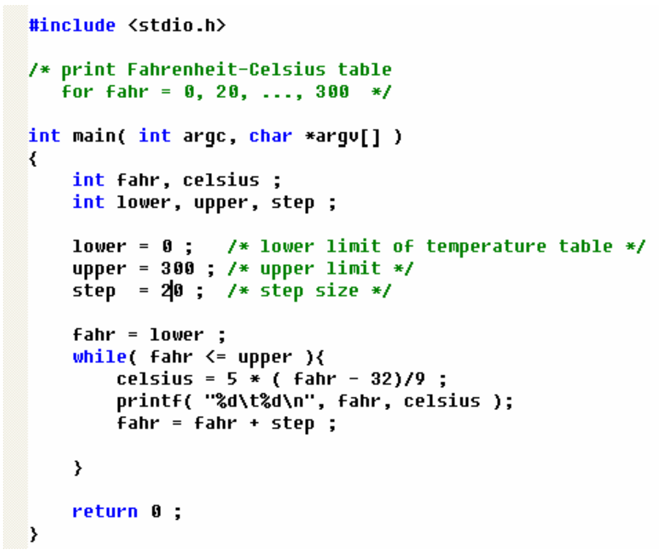


```
#include <stdio.h>

#define  HELLO_STRING   "hello, world\n"

int main( int argc, char* argv[] )
{
    printf( HELLO_STRING );

    return 0 ;
}
```

Figure 2: replace string “hello, world\n” with a macro HELLO_STRING.

Question 1: why we need to define a macro HELLO_STRING? Is printf("hello, world") not good? Think about pro & con (優缺點) of macro substitution.

Exercise 4: read section 1.2 from page 8 to page 14. Use visual studio to write another program in page 9, (project name is Fahrenheit_Celsius and source file is main.cpp)



```
#include <stdio.h>

/* print Fahrenheit-Celsius table
   for fahr = 0, 20, ..., 300  */

int main( int argc, char *argv[] )
{
    int fahr, celsius ;
    int lower, upper, step ;

    lower = 0 ;   /* lower limit of temperature table */
    upper = 300 ; /* upper limit */
    step  = 20 ;  /* step size */

    fahr = lower ;
    while( fahr <= upper ){
        celsius = 5 * ( fahr - 32)/9 ;
        printf( "%d\t%d\n", fahr, celsius );
        fahr = fahr + step ;

    }

    return 0 ;
}
```

Figure 3: program about Fahrenheit-Celsius table in page 9 of textbook.

Check your execution result, is it the same as that in page 8 ?

Exercise 5 (operations in remote machine):

(1) connect to remote machne 140.114.34.214 or 140.114.34.216 via SSH

(2) create directory **course** under your home directory

(3) upload directory **HelloWrold** (you create it in **Exercise 1**) to /home/[your home directory]/course via sftp

(4) Compile **main.cpp** into executable file **a.out** by icpc and gcc

(5) Change your password