

C Programming

Day22.B

2017.12.13

Preprocessing

Copyright (c) 2015 - 2017 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

```

#include <stdarg.h>
#include <stdio.h>

// va_list
// va_start
// va_arg
// va_end

// average ( 3, 12.2, 22.3, 4.5 ) ; // 3 more arg_list
// arg_list : 12.2, 22.3, 4.5
// num      :3
// average ( 5, 3.3, 2.2, 1.1, 5.5, 3.3 ) ; // 5 more arg_list
// arg_list : 3.3, 2.2, 1.1, 5.5, 3.3
// num      :5

double average ( int num, ... ) // num more arguments
{
    double sum = 0.0;
    int i;

    va_list arg_list; ////////////////////////////////////////////////////
    va_start ( arg_list, num ); ////////////////////////////////////////////////////

    for ( i = 0; i < num; i++ )
        sum += va_arg ( arg_list, double ); ////////////////////////////////////////////////////

    va_end ( arg_list ); ////////////////////////////////////////////////////

    return sum / num;
}

int main()
{
    double x, y;

    x = average ( 3, 12.2, 22.3, 4.5 ) ; // 3 more arg_list
    y = average ( 5, 3.3, 2.2, 1.1, 5.5, 3.3 ) ; // 5 more arg_list

    printf( "%f\n", x );
    printf( "%f\n", y );
}

```

macros defined

macro usage

extraction type

```
#include <stdarg.h>
#include <stdio.h>

// va_list
// va_start
// va_arg
// va_end

void argscan ( int num, ... ) // num more arguments
{
    int i;
    double x;
    char c;
    char *s;

    va_list arg_list; //////////////////////////////////////
    va_start ( arg_list, num ); //////////////////////////////////////

    i = va_arg ( arg_list, int ); //////////////////////////////////
    x = va_arg ( arg_list, double ); //////////////////////////////////
    // c = va_arg ( arg_list, int ); //////////////////////////////////
    // s = va_arg ( arg_list, int * ); //////////////////////////////////

    va_end ( arg_list ); //////////////////////////////////////

    printf("i= %d \n", i);
    printf("x= %f \n", x);
    // printf("c= %c \n", c);
    // printf("s= %s \n", s);
}

int main()
{
    argscan(4, 4, 4, 4, 10, 3.14, 'A', "hello" );
}
    i x c s
```

Something Wrong

```

#include <stdarg.h>
#include <stdio.h>

// va_list
// va_start
// va_arg
// va_end

void argscan ( int num, ... ) // num more arguments
{
    int i;
    double x;
    char c;
    char *s;

    va_list arg_list; //////////////////////////////////////
    va_start ( arg_list, num ); //////////////////////////////////////

    i = va_arg ( arg_list, int ); //////////////////////////////////////
    x = va_arg ( arg_list, double ); //////////////////////////////////////
    c = va_arg ( arg_list, int ); //////////////////////////////////////
    s = va_arg ( arg_list, char * ); //////////////////////////////////////

    va_end ( arg_list ); //////////////////////////////////////

    printf("i= %d \n", i);
    printf("x= %f \n", x);
    printf("c= %c \n", c);
    printf("s= %s \n", s);
}

int main()
{
    argscan( 4, 10, 3.14, 'A', "hello" );
}

```

to remove warnings

```

i= 10
x= 3.140000
c= A
s= hello

```

in a macro definition → make a string

```
#include <stdio.h>
```

a function-like macro

```
#define str( x ) #x  
#define str2( x ) "x"
```

```
int main(void) {
```

```
printf("%s %s\n", str(hello), str(world!));
```

```
printf("%s \n", str(hello) str(world!) );  
concatenated
```

```
printf("%s %s\n", str2(hello), str2(world!));
```

```
printf("%s \n", str2(hello) str2(world!));
```

```
}
```

```
hello world!  
helloworld!  
x x  
xx
```

in a macro \Rightarrow concatenate

```
#include <stdio.h>
```

```
#define f( x, y ) x##y
```

```
int main(void) {
```

```
    int i = 9999;
```

```
    int i0 = 20;
```

```
    int i1 = 30;
```

```
    int i2 = 40;
```

```
    printf("i= %d \n", i);
```

```
    printf("f(i, 0)= %d \n", f(i, 0));
```

```
    printf("f(i, 1)= %d \n", f(i, 1));
```

```
    printf("f(i, 2)= %d \n", f(i, 2));
```

```
}
```



```
#include <stdio.h>
```

```
int main(void) {
```

```
    printf("hello, \"John\\\"\\n");  
    printf("100%% \\n");
```

```
    printf("s1 \" \"s2 \" \"s3\" \"\\n");  
    printf("%s%s%s%s", "s1 \"\", \"s2 \"\", \"s3\", \"\\n");
```

```
    printf("  __LINE__  = %d \\n",  __LINE__ );  
    printf("  __FILE__  = %s \\n",  __FILE__ );  
    printf("  __DATE__  = %s \\n",  __DATE__ );  
    printf("  __TIME__  = %s \\n",  __TIME__ );  
    printf("  __STDC__  = %d \\n",  __STDC__ );  
    printf("  __FILE__  __DATE__  __TIME__  \"\\n");
```

```
}
```

```
hello, "John"  
100%  
s1 s2 s3  
s1 s2 s3  
  __LINE__  = 13  
  __FILE__  = t.c  
  __DATE__  = Dec 13 2017  
  __TIME__  = 23:07:54  
  __STDC__  = 1  
t.cDec 13 201723:07:54
```