

# C Programming

## Day12.B

2017.10.20

Unsigned Numbers,  
Double Pointers,

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 <stdio.h>

int main(void) {
    unsigned int u;
    int i;

    u = 0xffffffff;
    i = 0xffffffff;

    printf("unsigned u %%u= %u \n", u);
    printf("unsigned u %%d= %d *\n", u);
    printf("unsigned u %%x= %x \n", u);

    printf(" signed i %%u= %u *\n", i);
    printf(" signed i %%d= %d \n", i);
    printf(" signed i %%x= %x \n", i);

    printf("(u>0) = %d \n", (u>0));
    printf("(i>0) = %d \n", (i>0));
}

```

```

unsigned u %%u= 4294967295 print
unsigned u %%d= -1 *
unsigned u %%x= ffffffff print
 signed i %%u= 4294967295 *
 signed i %%d= -1
 signed i %%x= ffffffff
(u>0) = 1
(i>0) = 0

```

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

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

```
    unsigned char u;  
    char i;
```

→ 4-byte integer promotion

```
    u = 0xff; → 0x000000ff
```

```
    i = 0xff; → 0xffffffff
```

```
    printf("unsigned u %%u= %u \n", u);
```

```
    printf("unsigned u %%d= %d *\n", u);
```

```
    printf("unsigned u %%x= %x \n", u);
```

```
    printf(" signed i %%u= %u *\n", i);
```

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

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

```
    printf("(u>0) = %d \n", (u>0));
```

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

```
    }
```

```
unsigned u %u= 255  
unsigned u %d= 255 *  
unsigned u %x= ff  
 signed i %u= 4294967295 *  
 signed i %d= -1  
 signed i %x= ffffffff  
(u>0) = 1  
(i>0) = 0
```

## variables a, p, q

int a;

8a a

int \*p;

8p p

int \*\*q;

8q q

## initialized variables a, p, q

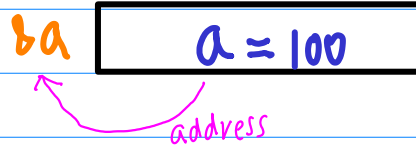
int a = 100;      &a      a = 100

int \*p = &a;      &p      p = &a

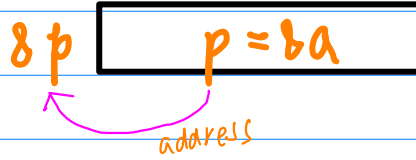
int \*\*q = &p;      &q      q = &p

# Operator &

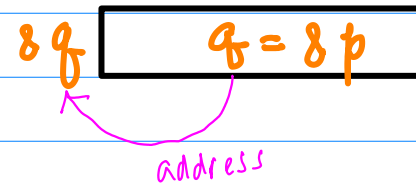
int a = 100;



int \*p = &a;

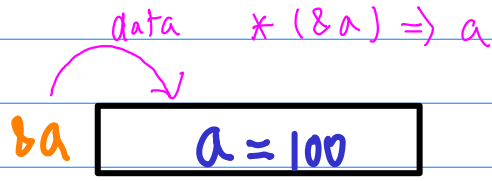


int \*\*q = &p;

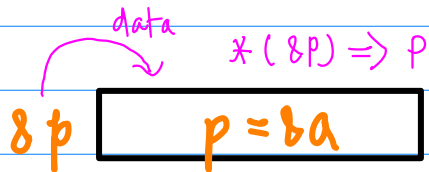


# Operator \*

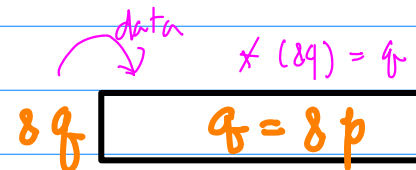
int a = 100;



int \*p = &a;



int \*\*q = &p;



# p's view point

int a = 100;

ba a = 100

int \*p = &a;

bp p = &a

int \*\*q = &p;

fq q = &p

int a = 100;

p \*p ≡ a = 100

\*p ≡ a

int \*p = &a;

bp p

int \*\*q = &p;

fq q = &p

# q's view point

int a = 100;

ba a = 100

int \*p = &a;

bp p = &a

int \*\*q = &p;

bq q = &p

int a = 100;

p a = 100

int \*p = &a;

b \*p = &a

int \*\*q = &p;

bq q

int a = 100;

\*q \*\*q = \*p = a = 100

\*\*q ≡ a

int \*p = &a;

b \*p = &a

\*p ≡ p

int \*\*q = &p;

bq q



int a = 100;

\*q



\*\*q

int \*p = &a;

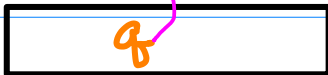
&



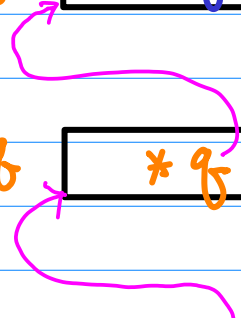
\*q

int \*\*q = &p;

&q



q



# pass-by-reference of an int variables

int a = 100;

$\&a$



int b = 200;

$\&b$



Swap

swap (  $\&a$  ,  $\&b$  );

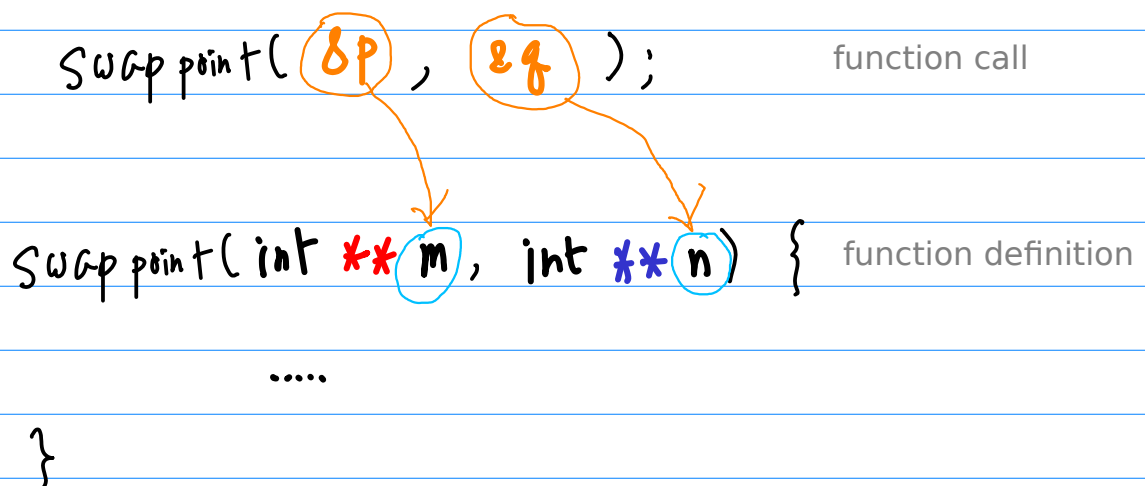
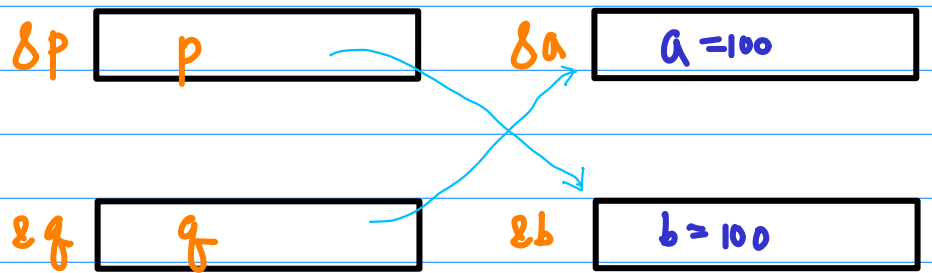
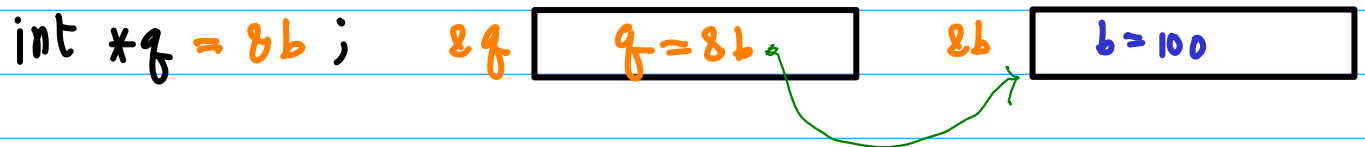
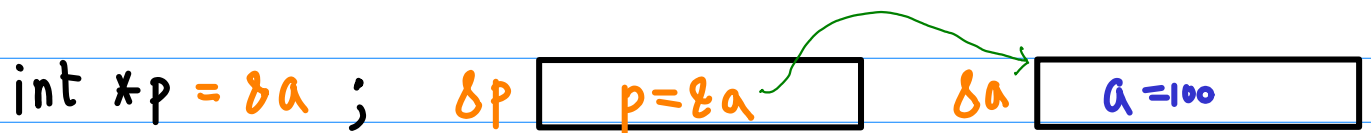
function call

swap ( int \*m , int \*n ) {

function definition

}

# pass-by-reference of an int \* variables



conversion ○  $\xrightarrow{\text{to}}$  ○

str to d

string  $\rightarrow$  double

str to l

string  $\rightarrow$  long

str to ul

string  $\rightarrow$  unsigned long

→  
" AAA 3.14 BBB 0.3333 ccc "

↙ strtod  
3.14

str  
str+3  
ptr  
ptr+4  
ptr  
" AAA 3.14 BBB 0.333 ccc "

3.14 ← strtod ( str + 3 , & ptr )

0.333000 ← strtod ( ptr + 4 , & ptr ) .

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

int main(void) {
    char *str = "AAA 3.14 BBB 0.333CCC";
    char *ptr;
    double x;

    printf("str= %s \n", str);

    x = strtod(str+3, &ptr);

    printf("x= %f \n", x);

    printf("ptr= %s \n", ptr);

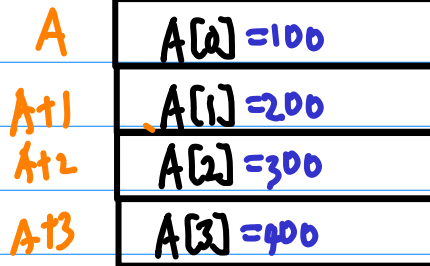
    x = strtod(ptr+4, &ptr);

    printf("x= %f \n", x);
    printf("ptr= %s \n", ptr);
}
```

```
str= AAA 3.14 BBB 0.333CCC
x= 3.140000
ptr= BBB 0.333CCC
x= 0.333000
ptr= CCC
```

# an array of pointers

int A[4];



int \*B[4];

