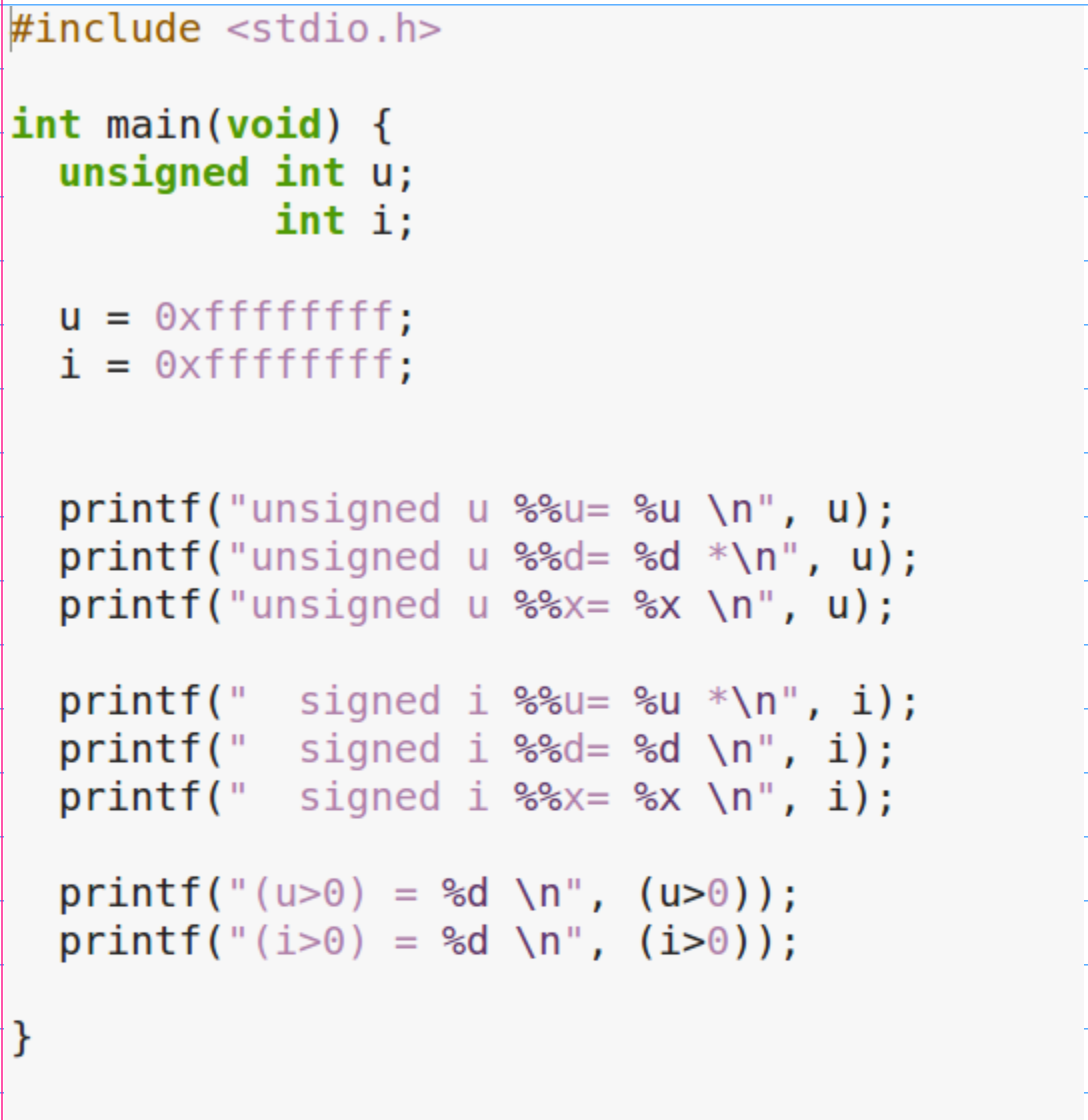
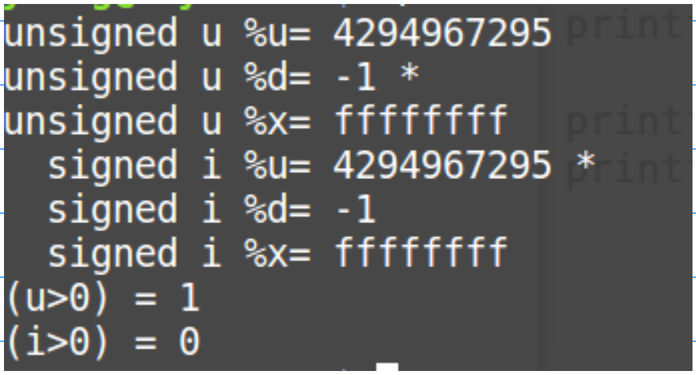
C Programming
Day12.B

2017.10.20

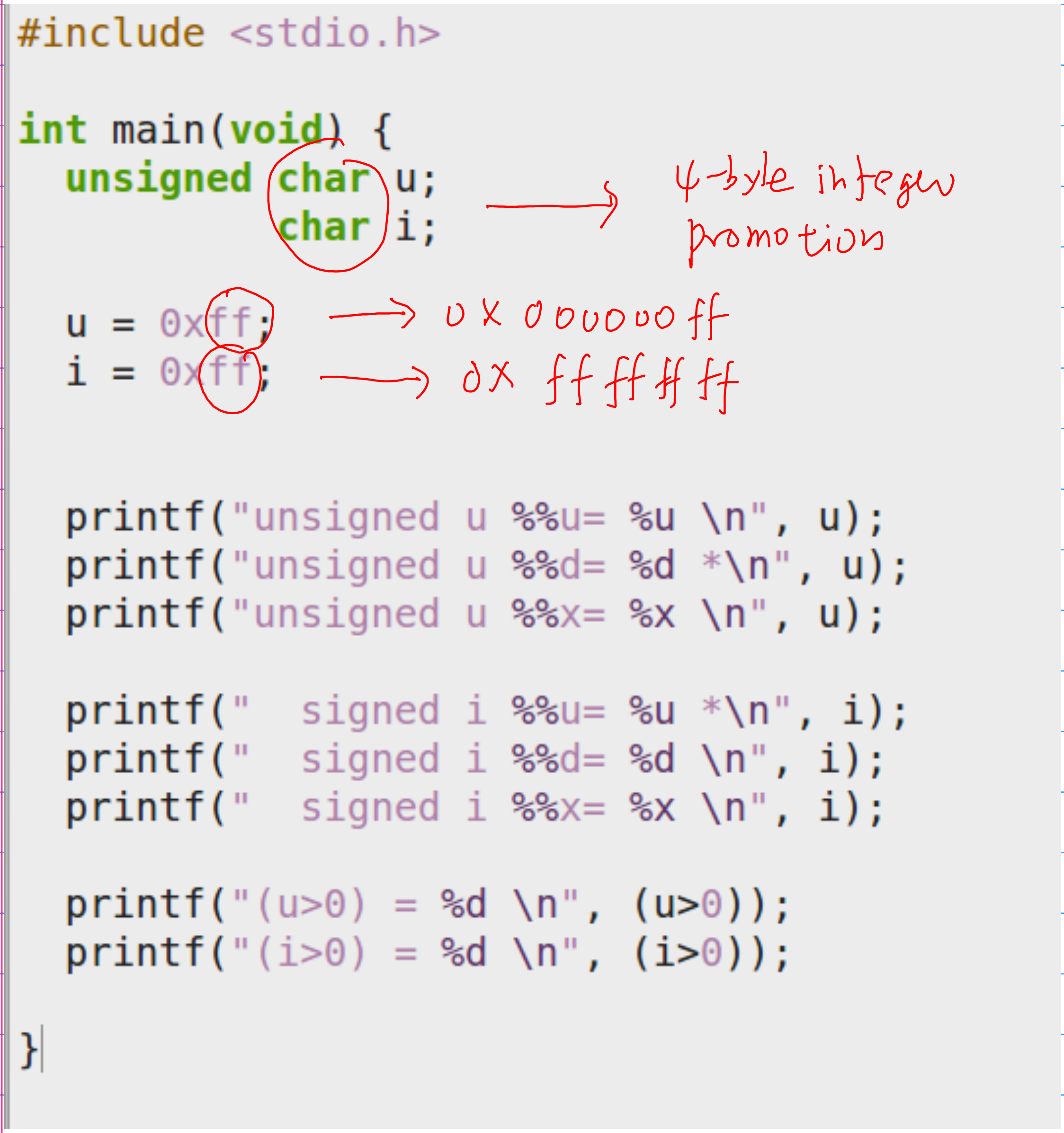
Unsigned Numbers,
Double Pointers,



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

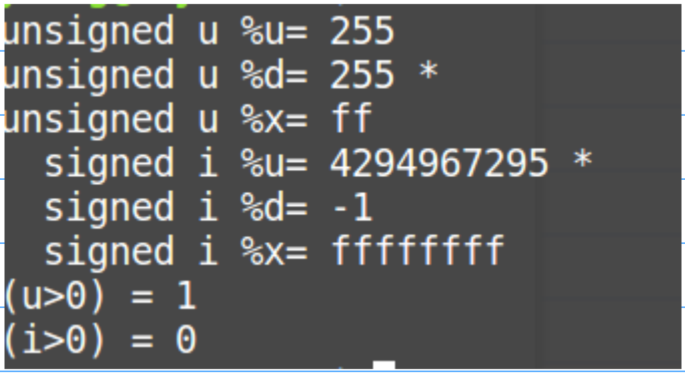


```c
#include <stdio.h>

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

  u = 0xff;
  i = 0xff;


  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));

}
```

→ 4-byte integer promotion

→ 0x 000000ff

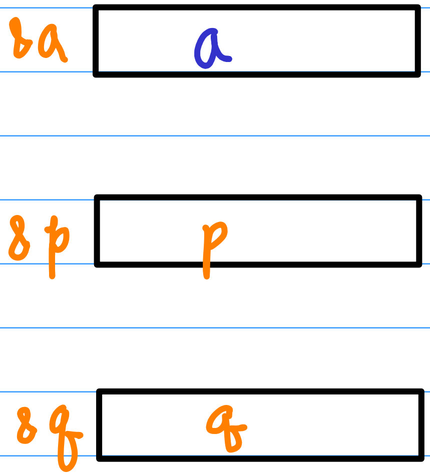
→ 0x ffffffff



```
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
```

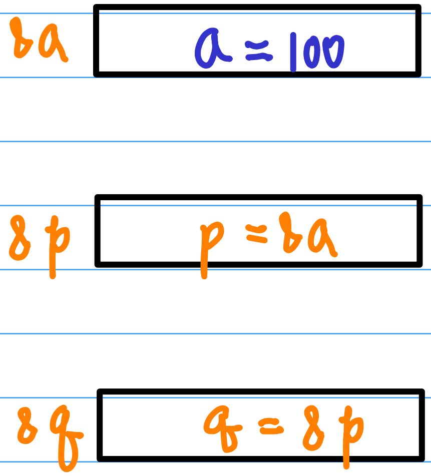
variables a, p, q

```
int a;

int *p;

int **q;
```



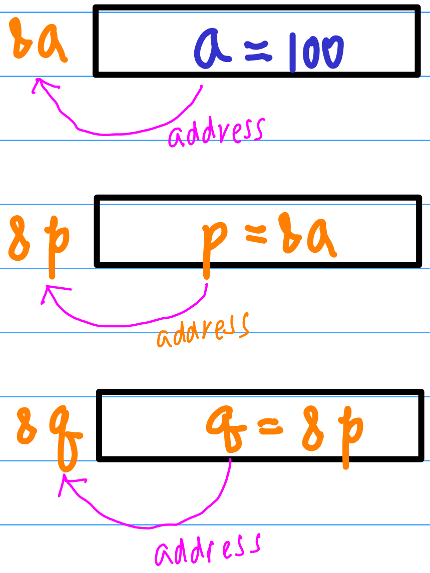
initialized variables a, p, q

```
int a = 100;

int *p = &a;

int **q = &p;
```



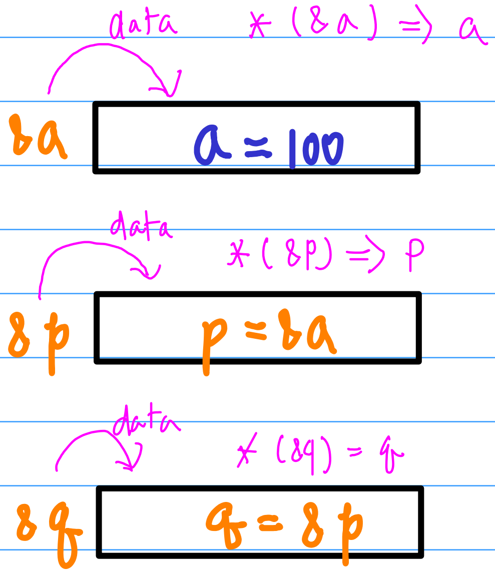
Operator &

```
int  a = 100;
int *p = &a;
int **q = &p;
```



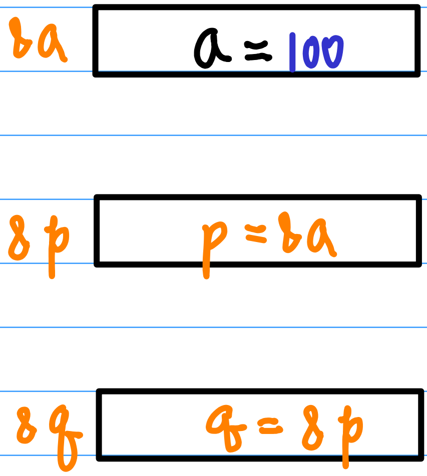
Operator *

```
int  a = 100;
int *p = &a;
int **q = &p;
```

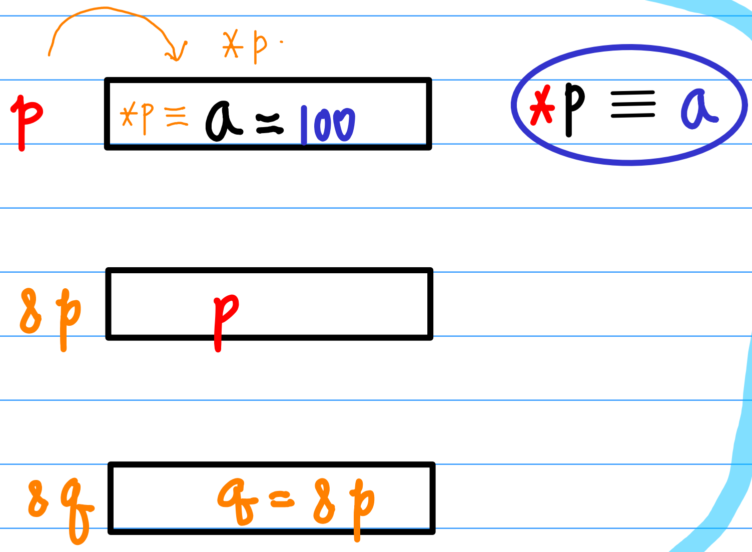


p's view point

```
int a = 100;
int *p = &a;
int **q = &p;
```

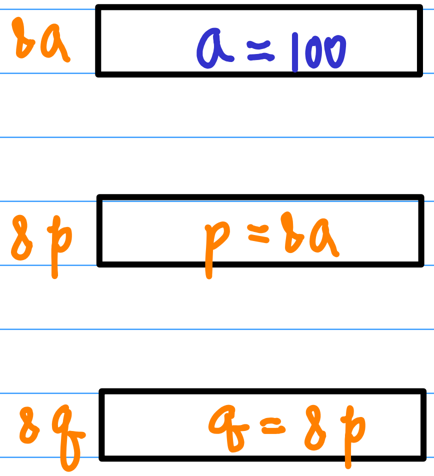


```
int a = 100;
int *p = &a;
int **q = &p;
```

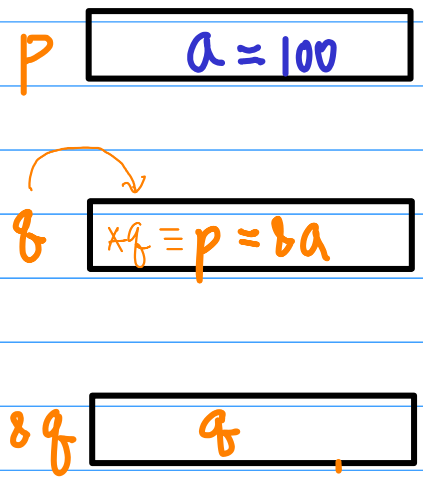


q's view point

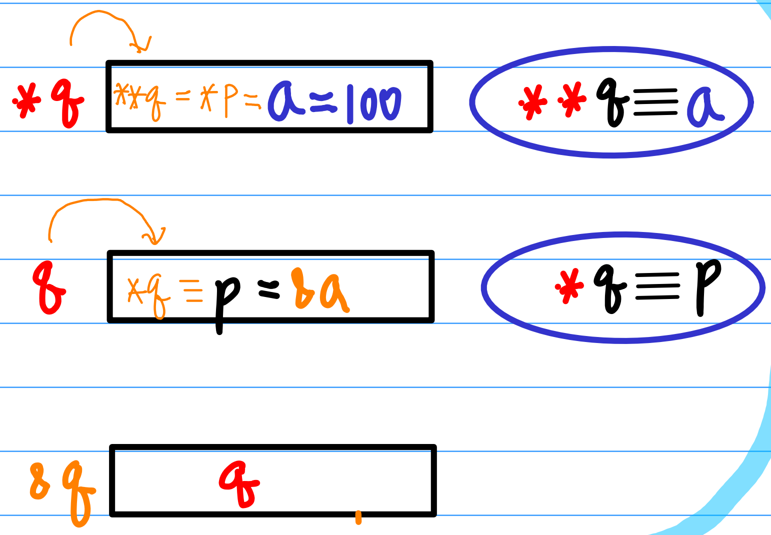
```
int a = 100;
int *p = &a;
int **q = &p;
```



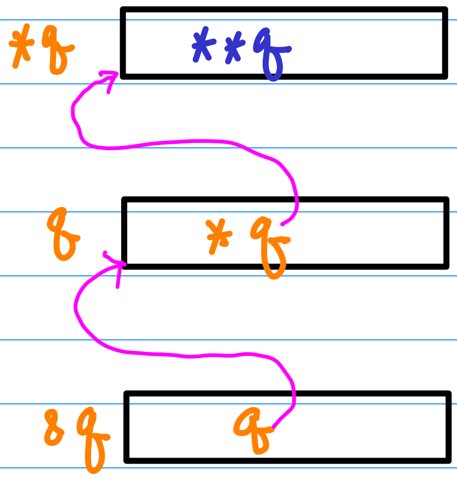
```
int a = 100;
int *p = &a;
int **q = &p;
```



```
int a = 100;
int *p = &a;
int **q = &p;
```

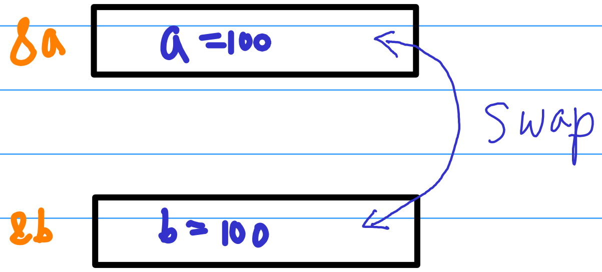


```
int   a  = 100;

int  *p = &a;

int **q  = &p;
```



pass-by-reference of an int variables

```
int a = 100;

int b =200 ;
```



```
swap ( &a , &b );
```

function call

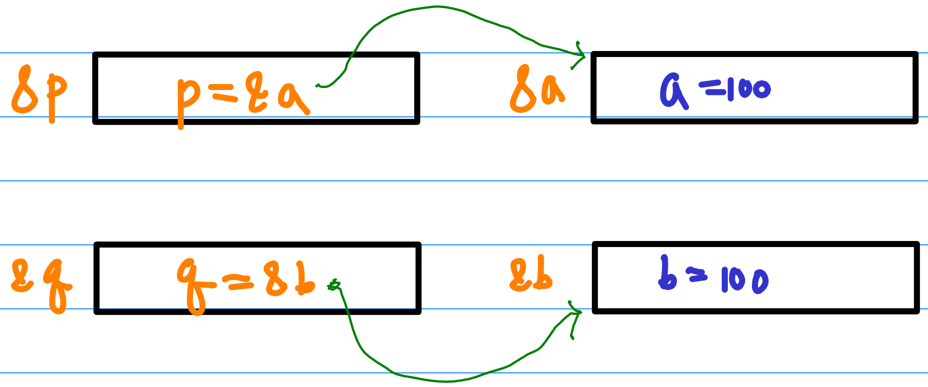
```
swap ( int *m, int *n) {

} .
```

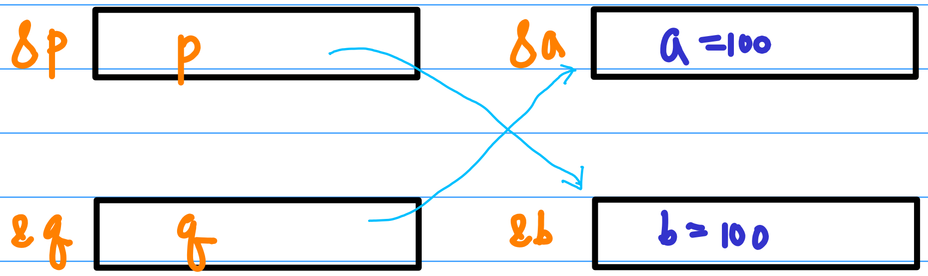
function definition

pass-by-reference of an int * variables

int *p = &a ;

int *q = &b ;

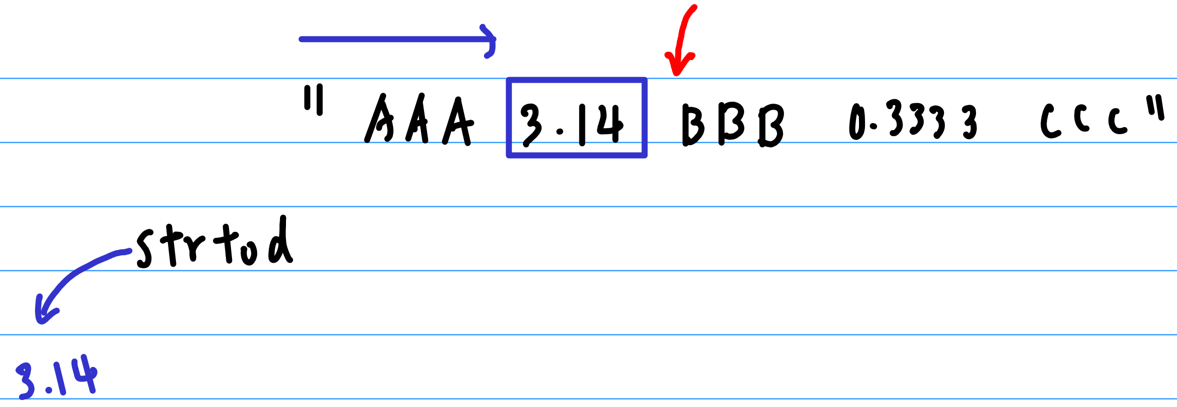


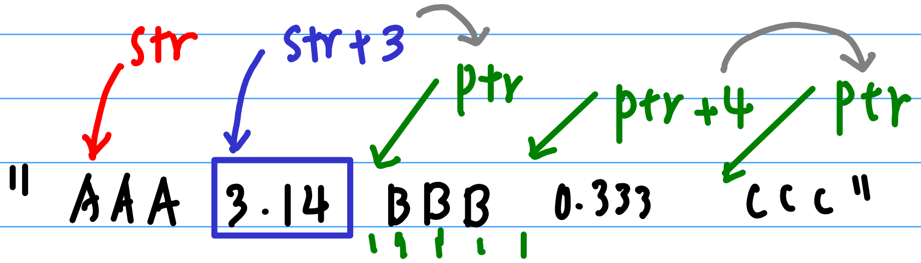


```
swappoint( &p , &q );                        function call

swappoint(int **m, int **n) {                function definition
        .....
}
```

conversion ○ $\xrightarrow{to}$ ○

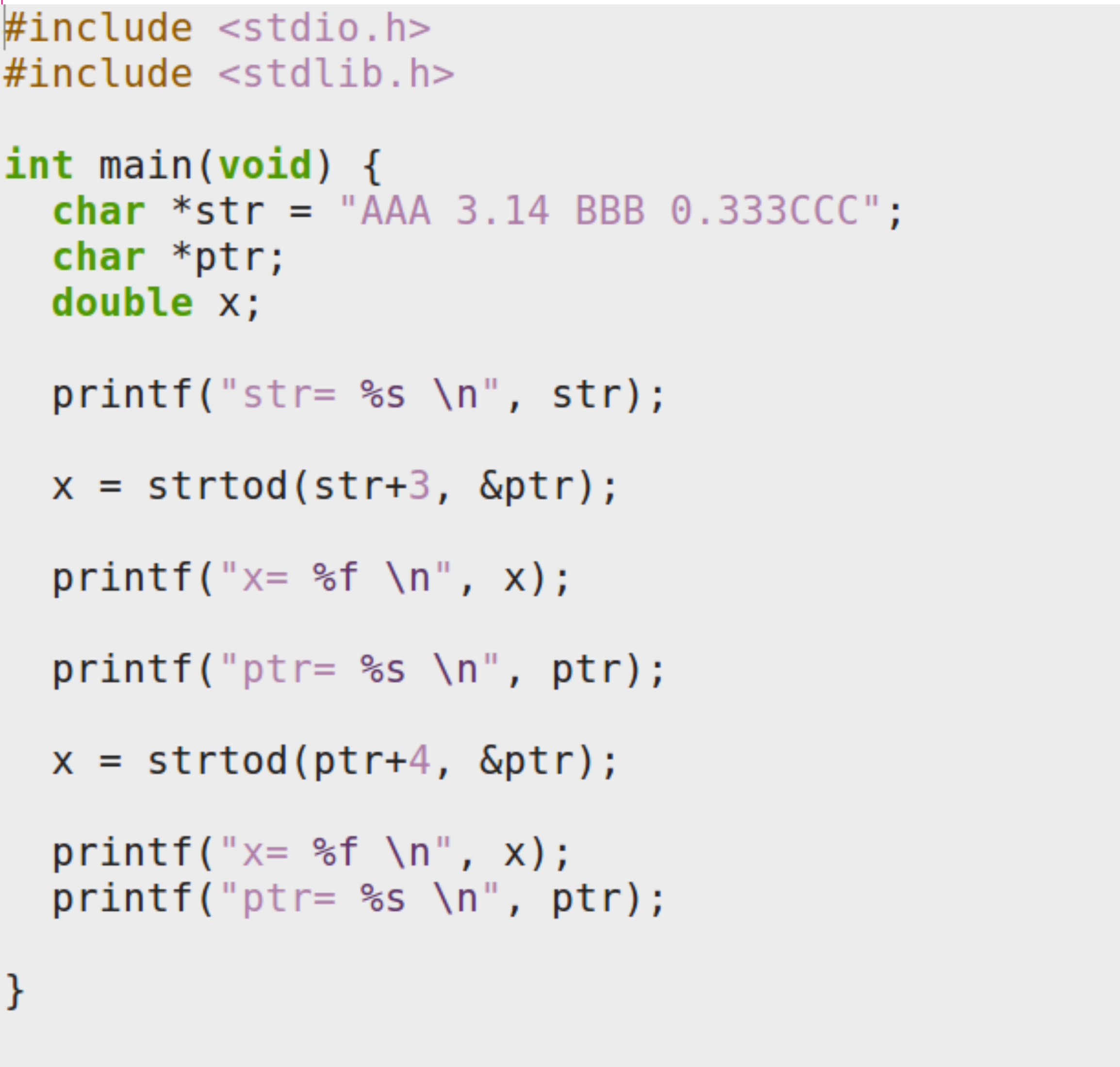
str to d	string → double
str to l	string → long
str to ul	string → unsigned lon g



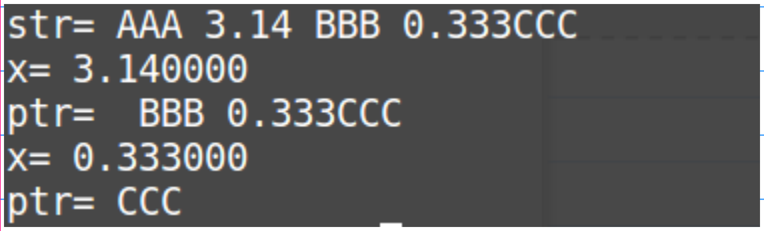


3.14 ← strtod(str+3, &ptr)

0.333000 ← strtod(ptr+4, &ptr)



```c
#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);

}
```

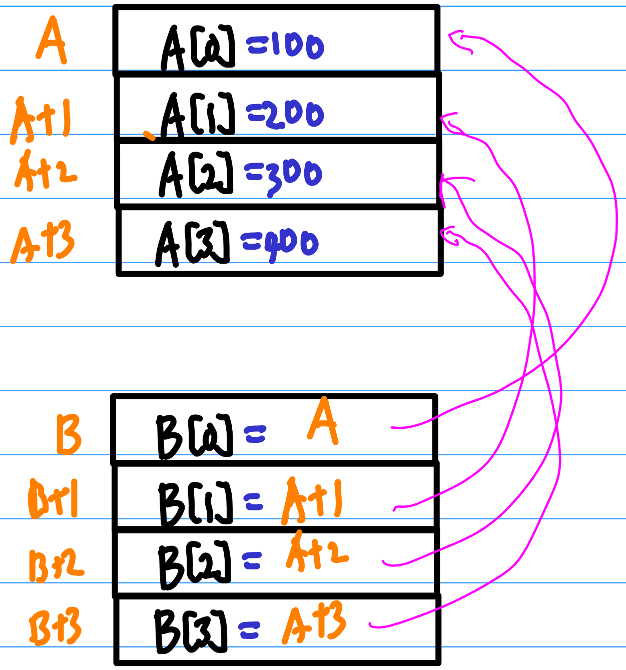


```
str= AAA 3.14 BBB 0.333CCC
x= 3.140000
ptr=  BBB 0.333CCC
x= 0.333000
ptr= CCC
```

an array of pointers

int A [4];

int *B [4];


A
A[0] =100
A+1
A[1] =200
A+2
A[2] =300
A+3
A[3] =400
B
B[0] = A
B+1
B[1] = A+1
B+2
B[2] = A+2
B+3
B[3] = A+3