

# About I/O

郭至軒 (KuoE0)

[KuoE0.tw@gmail.com](mailto:KuoE0.tw@gmail.com)

[KuoE0.ch](http://KuoE0.ch)





# Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

<http://creativecommons.org/licenses/by-sa/3.0/>

**Latest update: Feb 27, 2013**

# Standard Input & Output

```
KuoE0 at Chih-Hsuans-Mac-mini 0:00 Desktop/temp  
➤ ./a.out  
1 2  
3  
3 4  
7  
5 6  
11  
7 8  
15
```

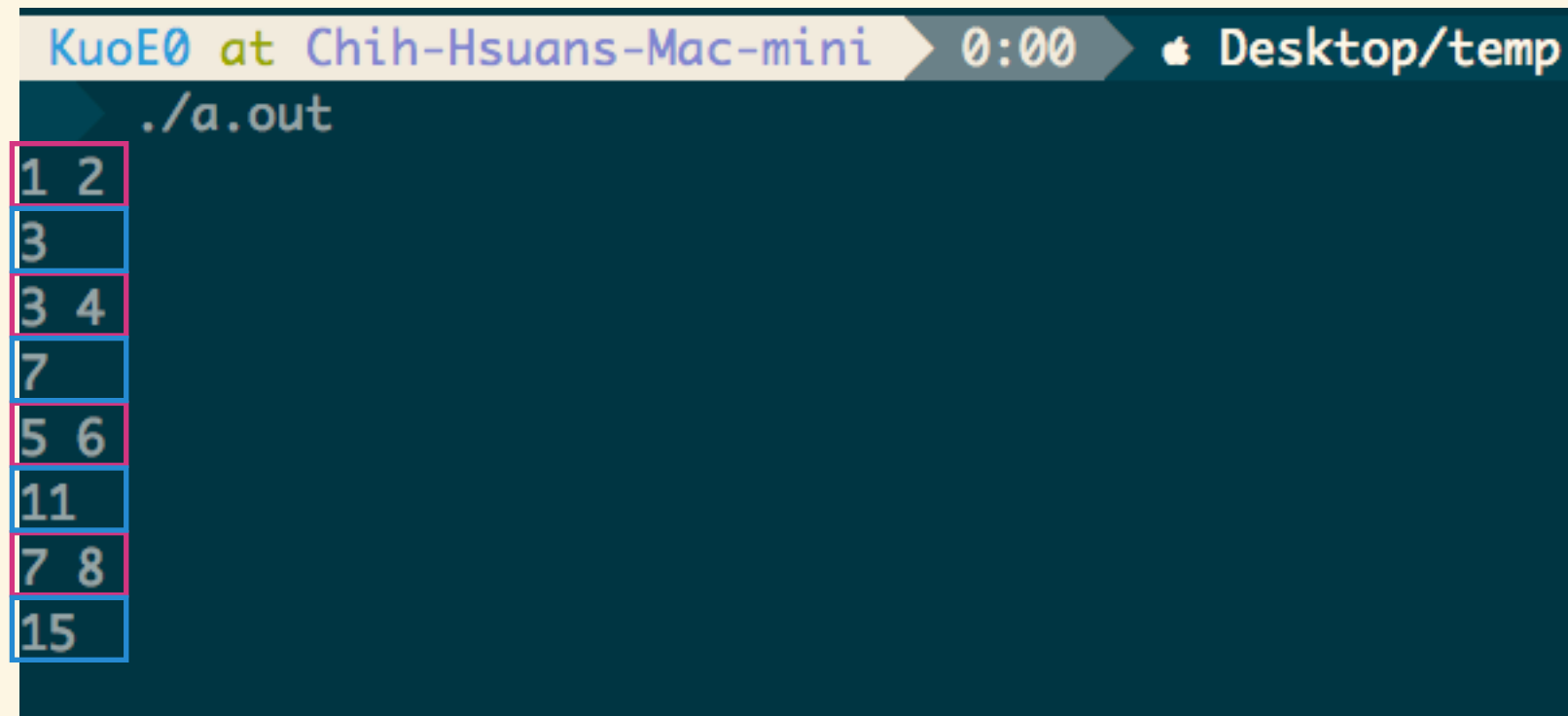
- 標準輸入
- 標準輸出

# Standard Input & Output

```
KuoE0 at Chih-Hsuans-Mac-mini 0:00 Desktop/temp  
./a.out  
1 2  
3  
3 4  
7  
5 6  
11  
7 8  
15
```

- 標準輸入
- 標準輸出

# Standard Input & Output



A terminal window titled "KuoE0 at Chih-Hsuans-Mac-mini" with a timer at "0:00" and the path "Desktop/temp". The prompt is `./a.out`. The input sequence is: `1 2`, `3`, `3 4`, `7`, `5 6`, `11`, `7 8`, and `15`. The output sequence is: `1 2`, `3`, `3 4`, `7`, `5 6`, `11`, `7 8`, and `15`. The input and output are interleaved, with the output matching the input line by line.

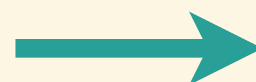
```
KuoE0 at Chih-Hsuans-Mac-mini 0:00 Desktop/temp
./a.out
1 2
3
3 4
7
5 6
11
7 8
15
```

- 標準輸入
- 標準輸出



標準輸入→由鍵盤輸入

標準輸出→由螢幕輸出



A rectangular icon with rounded corners, divided into two horizontal sections. The top section is white with horizontal grey lines and a vertical red margin line on the left. The bottom section is solid blue. The text 'input file' is written in blue in the top section, and 'TXT' is written in white in the bottom section.

input file

TXT

A rectangular icon with rounded corners, divided into two horizontal sections. The top section is white with horizontal grey lines and a vertical red margin line on the left. The bottom section is solid blue. The text 'output file' is written in blue in the top section, and 'TXT' is written in white in the bottom section.

output file

TXT

# Multiple Test Case

1 2

3 4

5 6

7 8



3

7

11

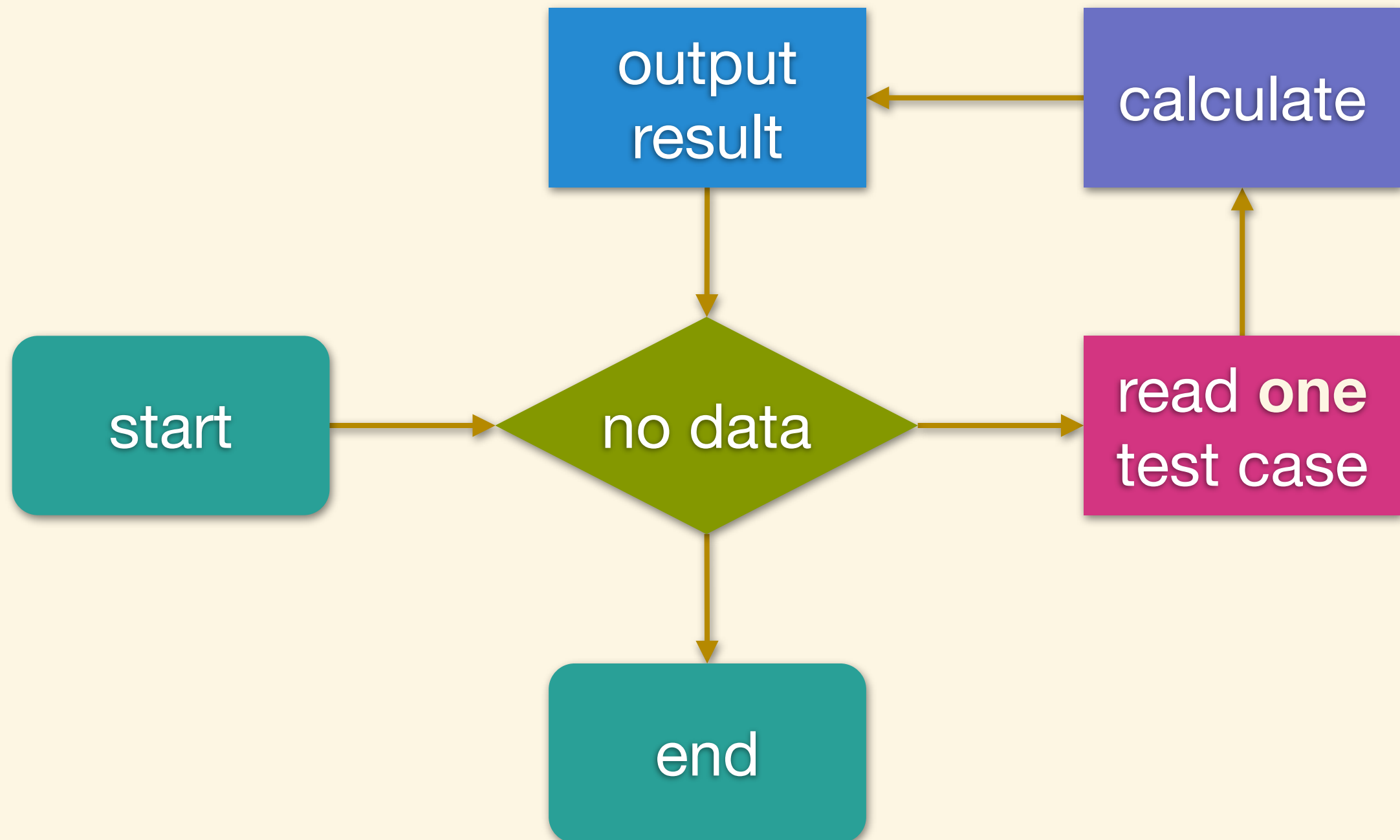
15

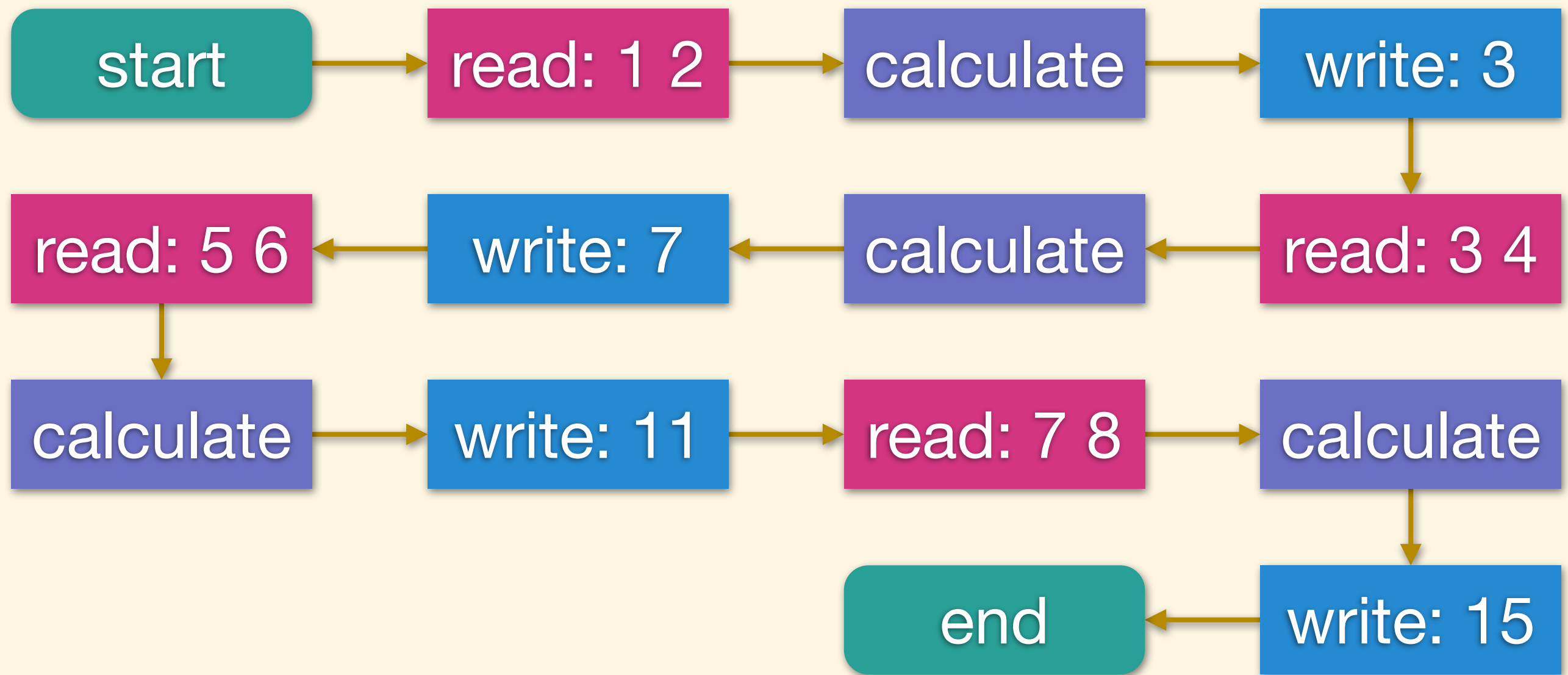
input file

output file



# Continuous Processing





# End Of File

若題目未指定測資終止條件， 則為判斷 EOF 作為終止條件！

scanf

```
while (scanf() != EOF)
{
    ...
}
```

fgets

```
while (fgets() != 0)
{
    ...
}
```

cin

```
while (cin >> x)
{
    ...
}
```

# File Input & Output

## fopen

```
FILE *in = fopen("inputfile");  
FILE *out =  
fopen("outputfile");  
  
fscanf(in, ...);  
fprintf(out, ...);  
  
fclose(in);  
fclose(out);
```

## fstream (C++ only)

```
ifstream in =  
ifstream("inputfile");  
  
ofstream out =  
ofstream("outputfile");  
  
in >> x;  
out << x;  
  
in.close();  
out.close();
```

If you write the code like  
previous page in contest, you  
will **lose** the contest.

# freopen

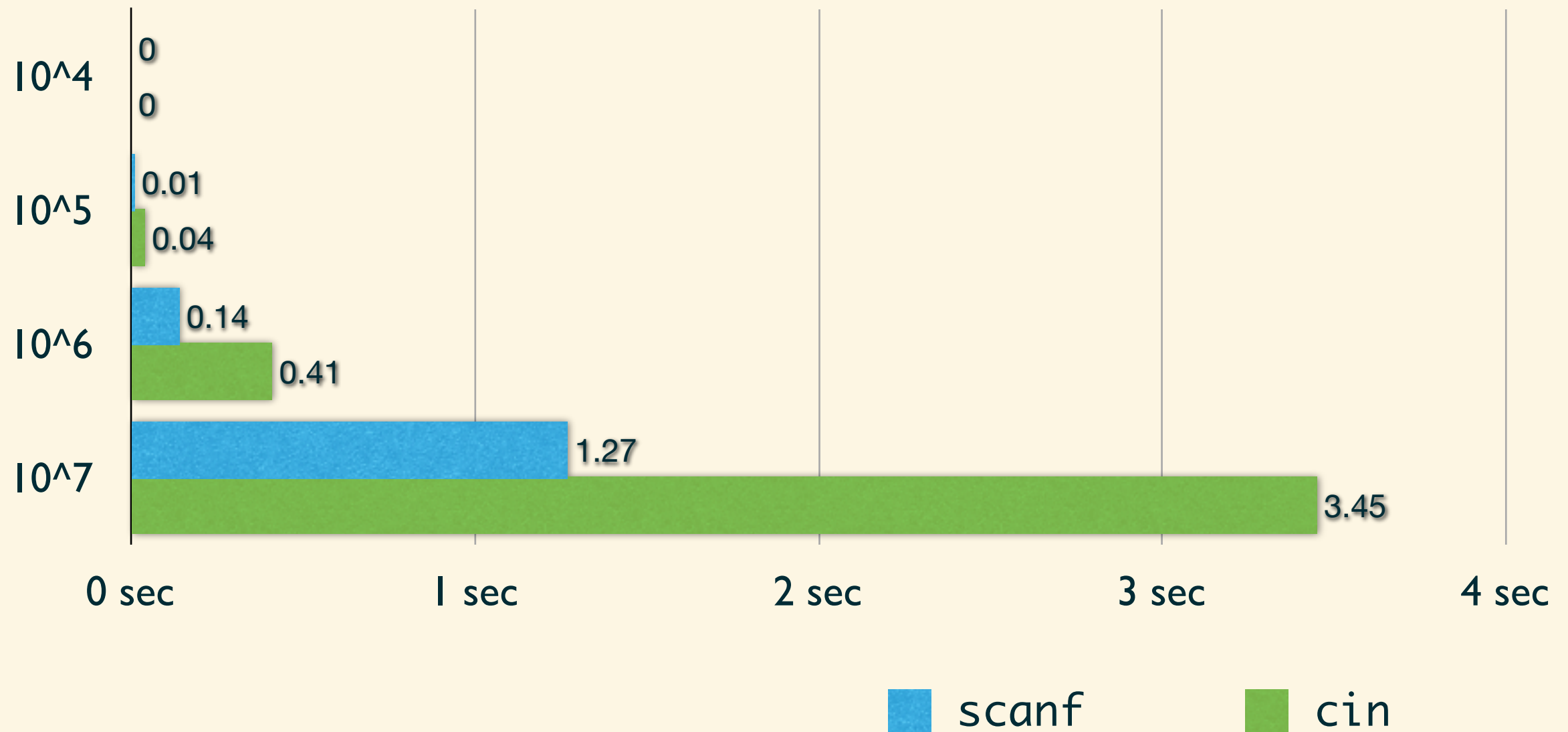
redirect the file I/O to standard I/O

```
freopen("inputfile", "r", stdin);  
freopen("output", "w", stdout);
```

```
scanf(...);  
printf(...);
```

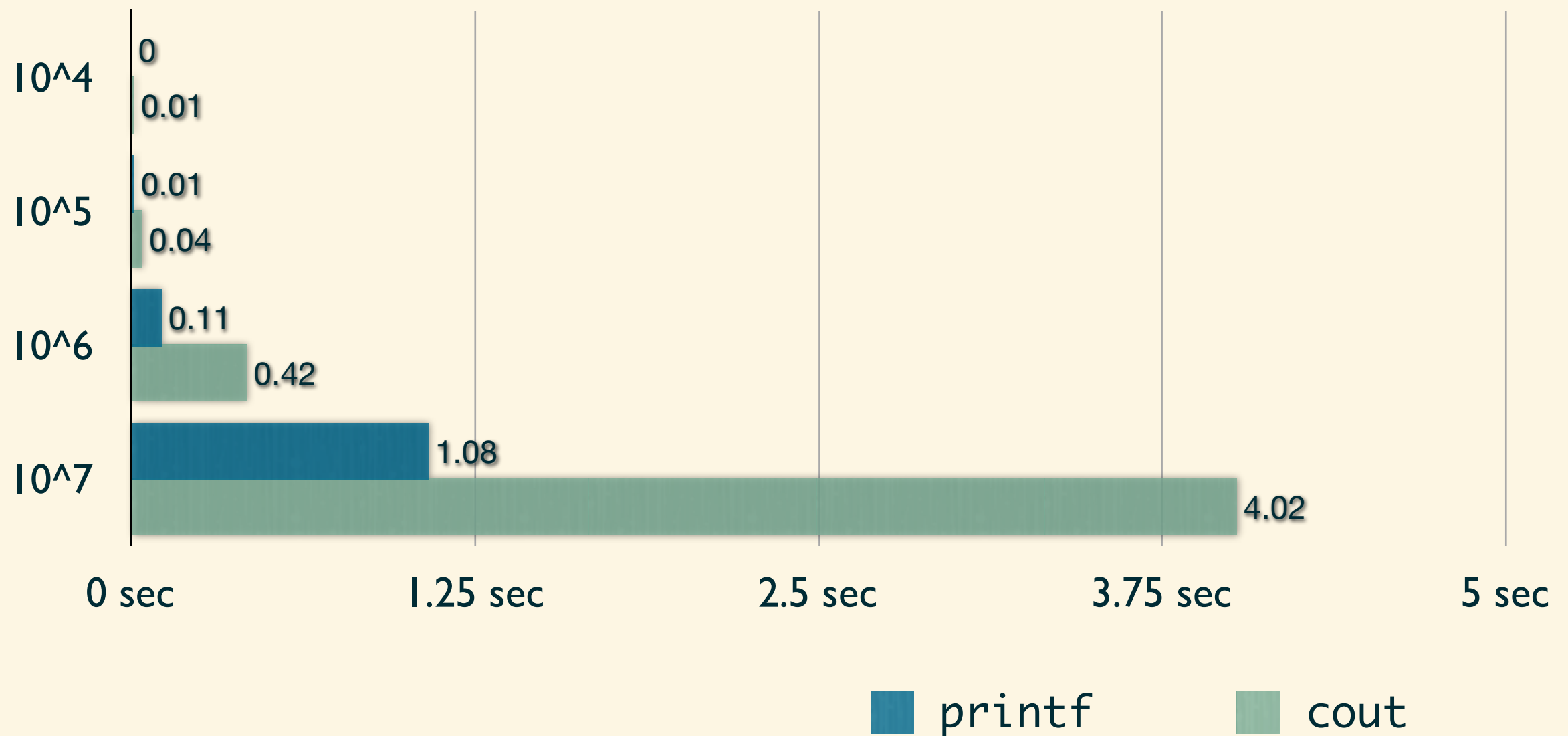
# I/O Performance

# scanf & cin





# printf & cout



# Improve Stream I/O

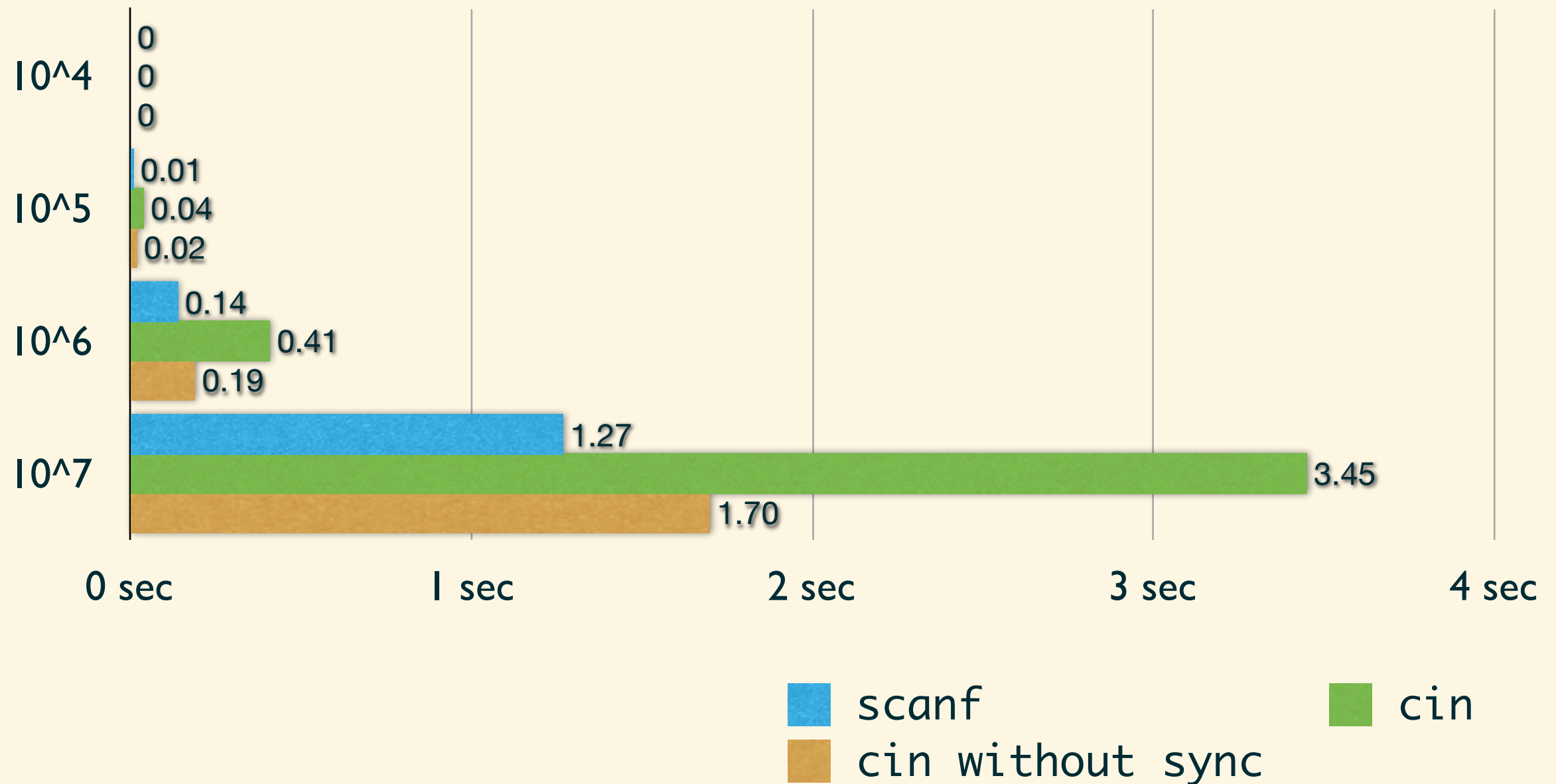
Stream I/O need to keep themselves in sync with the underlying C library.

Add this line in code,

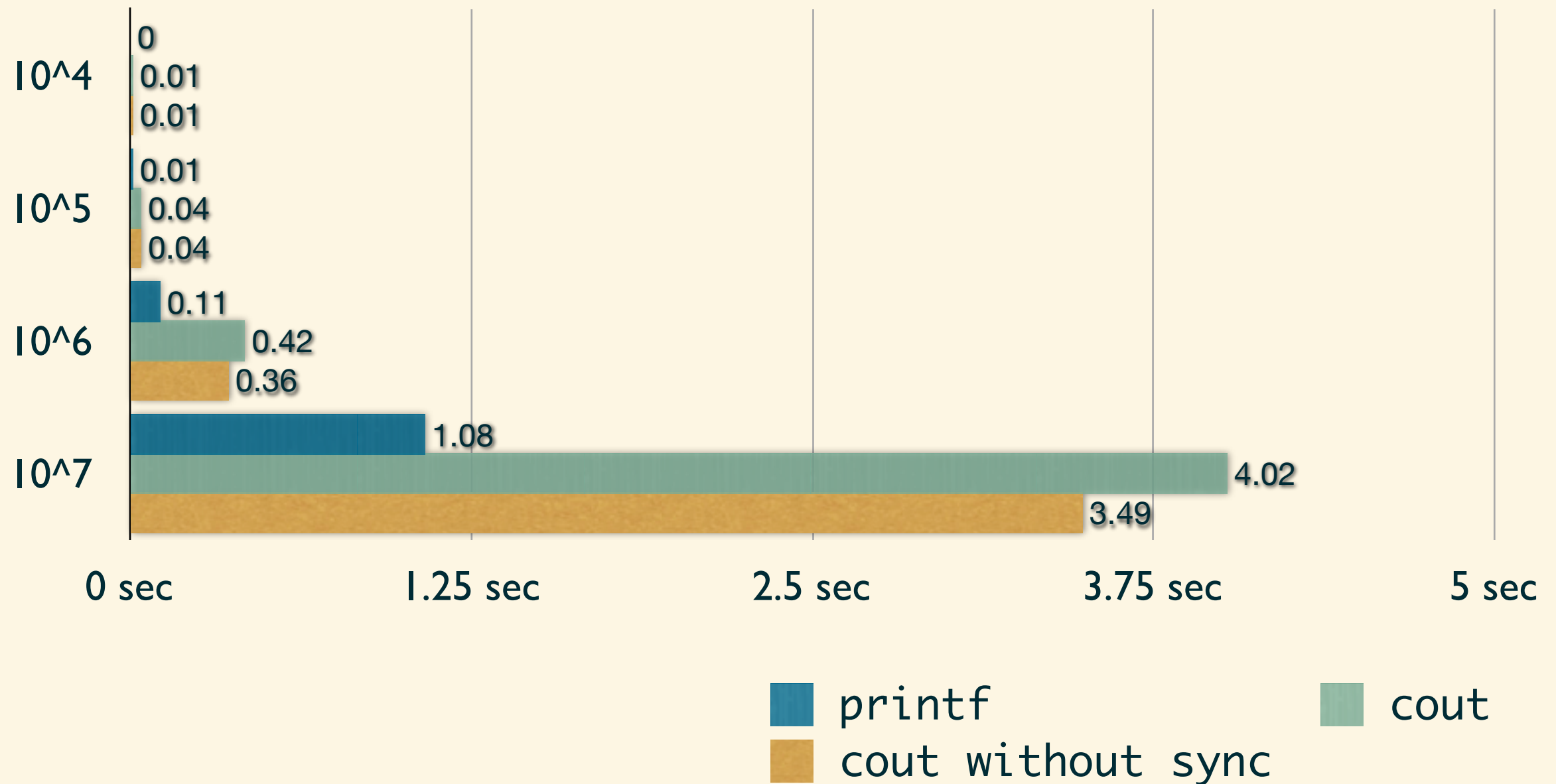
```
std::ios::sync_with_stdio(false);
```

if you won't use C I/O.

# cin without sync



# cout without sync



Stream I/O is Still Slow

Recommend  
to use C I/O.

# Buffered Technique

Base on fgets/puts function.

fgets/puts are **faster** than scanf/printf.

# Buffered Read

Read mass data, and parse by self.

```
char buf[ 1000 ];  
int a, b;  
fgets( buf, sizeof( buf ), stdin );  
sscanf( buf, "%d %d", &a, &b );
```

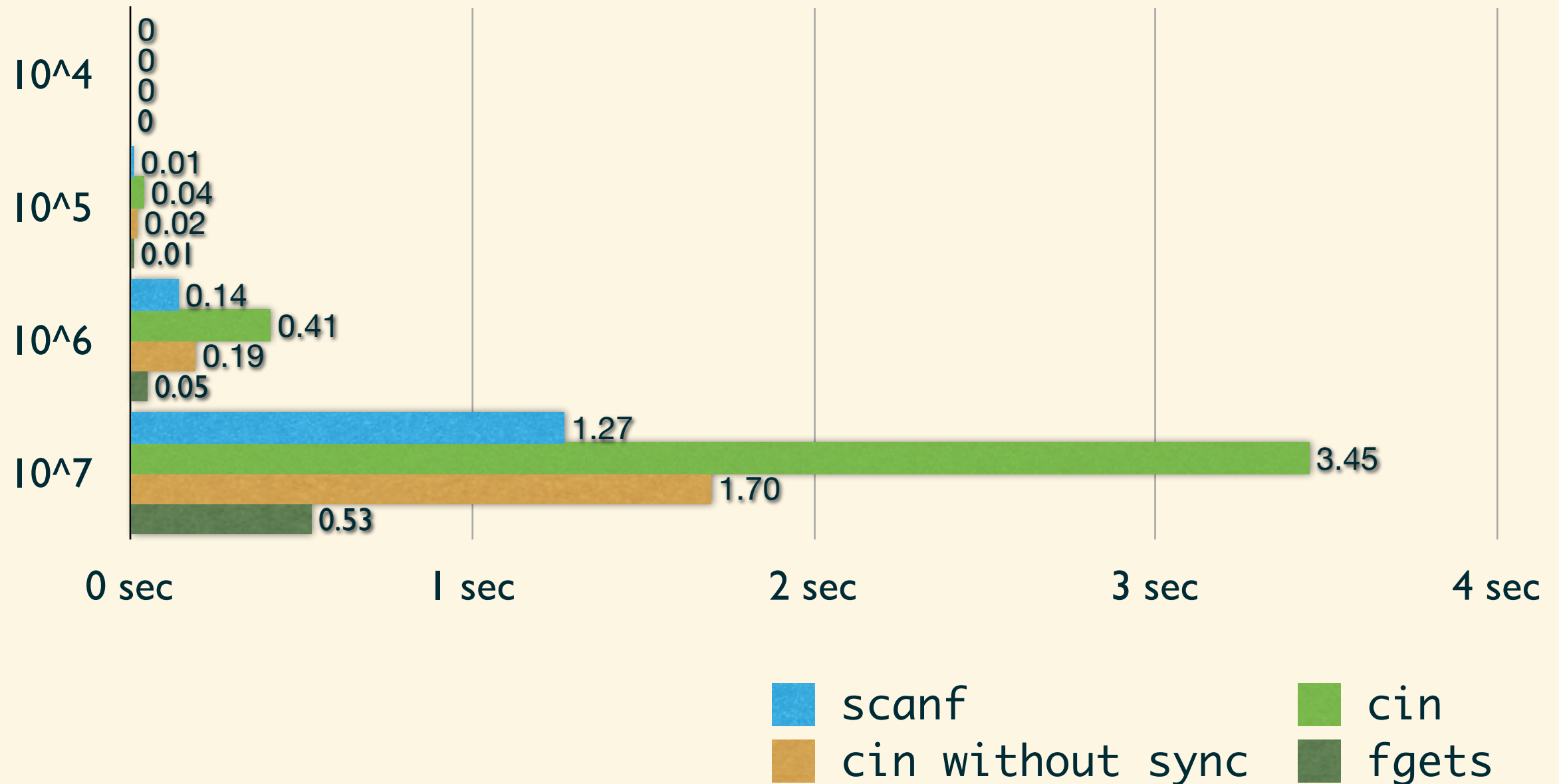
# Buffered Write

Store mass data into temporal buffer,  
and write them once.

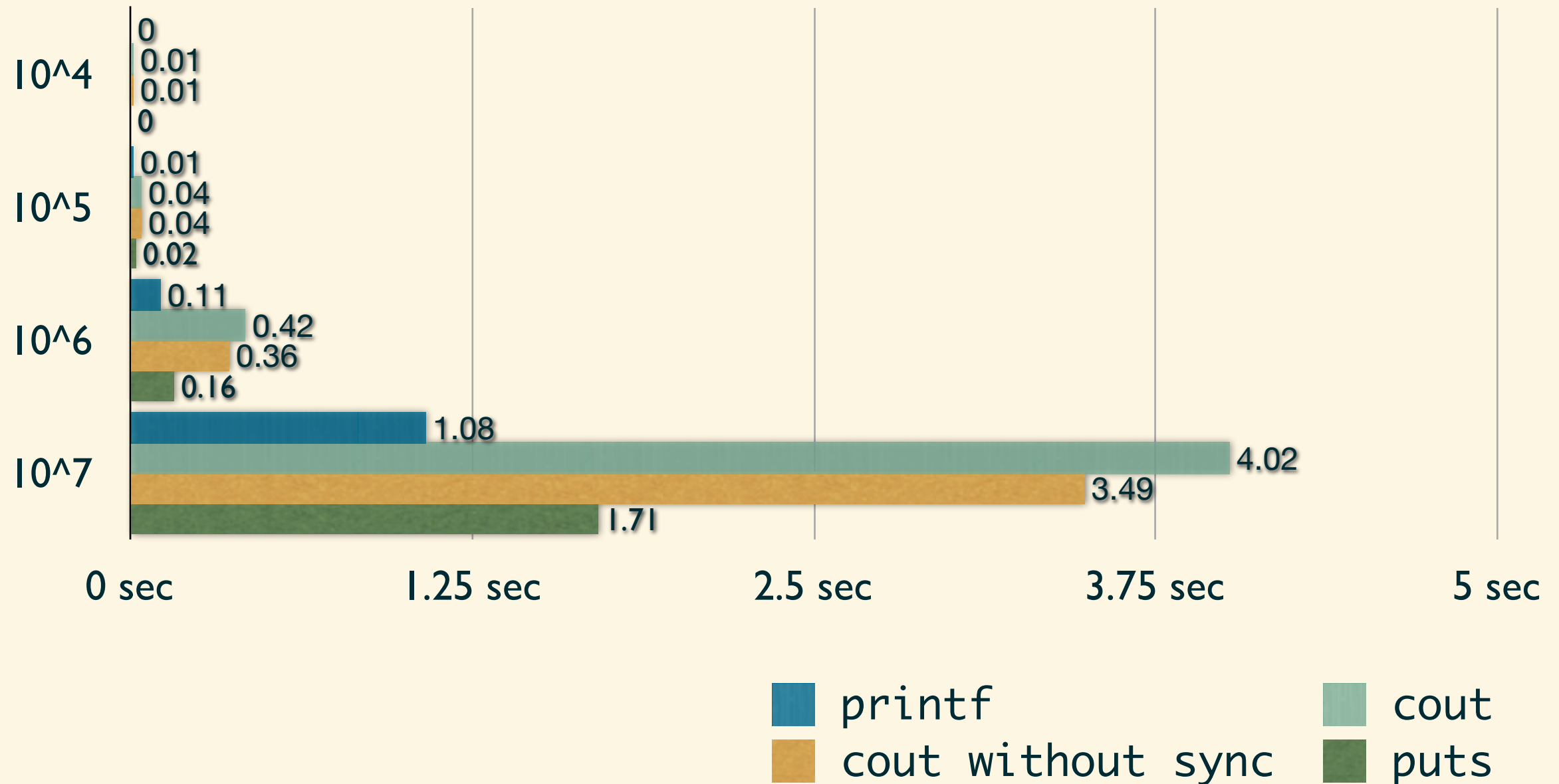
```
char buf[ 1000 ];  
int ret = a + b;  
sprintf( buf, "%d", ret );  
puts( buf );
```



# Buffering Technique



# Buffering Technique



# Advanced Parsing Skill

**strtok** Split string into tokens.

```
char* strtok( char *str, const char *delimiters );
```

str: 欲切割之字串

delimiters: 分隔字符字串

return value: 指向當前切割字串之指標, 若切割完畢則回傳 NULL

# strtok

original string:

A “corpus” is a collection of texts of written (or spoken) language presented in electronic form.

take out all words:

A	corpus	is	a
collection	of	texts	of
written	or	spoken	language
presented	in	electronic	form

`char` str = “A \“corpus\” is a collection of texts of written (or spoken) language presented in electronic form.”

```
for ( char *token = strtok( str, “ \”().” ); token !=  
NULL; token = strtok( NULL, “ \”().” ) ) {  
    puts( token );  
}
```

delimiters: “ \”().”



start position



global pointer



A

\“

c

o

r

p

u

s

\”

i

s

a

c

o

l

l



e

c

t

i

o

n

o

f

t

e

x

t

s

o

f

w

r

e

t

t

e

n

(

o

r

s

p

o

k

e

n

)

l

a

n

g

u

a

g

e

p

r

e

s

e

n

t

e

d

i

n

e

l

e

t

r

o

n

i

c

f

o

r

m

.

\0

delimiters: “ \”().”



start position



global pointer



A

\0

\“

c

o

r

p

u

s

\”

i

s

a

c

o

l

l



e

c

t

i

o

n

o

f

t

e

x

t

s

o

f

w

r

e

t

t

e

n

(

o

r

s

p

o

k

e

n

)

l

a

n

g

u

a

g

e

p

r

e

s

e

n

t

e

d

i

n

e

l

e

t

r

o

n

i

c

f

o

r

m

.

\0

Return address of start position.

“A”



delimiters: “ \”().”



start position



global pointer



A	\0	\“	c	o	r	p	u	s	\”		i	s		a		c	o	l	l
---	----	----	---	---	---	---	---	---	----	--	---	---	--	---	--	---	---	---	---

e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----

delimiters: “ \”().”



start position



global pointer



A	\0	\0	c	o	r	p	u	s	\0		i	s		a		c	o	l	l
---	----	----	---	---	---	---	---	---	----	--	---	---	--	---	--	---	---	---	---



e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----

Return address of start position.

“corpus”

delimiters: “ \”().”



start position



global pointer



A	\0	\0	c	o	r	p	u	s	\0		i	s		a		c	o	l	l
---	----	----	---	---	---	---	---	---	----	--	---	---	--	---	--	---	---	---	---

e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----

delimiters: “ \”().”



start position

global pointer

A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a		c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	--	---	---	---	---

e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----

Return address of start position.

“is”

delimiters: “ \”().”



start position



global pointer



A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a		c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	--	---	---	---	---

e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----

delimiters: “ \”().”



start position



global pointer



A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a	\0	c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	----	---	---	---	---

e	c	t	i	o	n		o	f		t	e	x	t	s		o	f		w
---	---	---	---	---	---	--	---	---	--	---	---	---	---	---	--	---	---	--	---

r	e	t	t	e	n		(	o	r		s	p	o	k	e	n	)		l
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---	---	---	--	---

a	n	g	u	a	g	e		p	r	e	s	e	n	t	e	d		i	n
---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	--	---	---

	e	l	e	t	r	o	n	i	c		f	o	r	m	.	\0
--	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	----



Return address of start position.

“a”

After 5 minutes...

delimiters: “ \”().”



start position



global pointer

A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a	\0	c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	----	---	---	---	---

e	c	t	i	o	n	\0	o	f	\0	t	e	x	t	s	\0	o	f	\0	w
---	---	---	---	---	---	----	---	---	----	---	---	---	---	---	----	---	---	----	---

r	e	t	t	e	n	\0	\0	o	r	\0	s	p	o	k	e	n	\0	\0	l
---	---	---	---	---	---	----	----	---	---	----	---	---	---	---	---	---	----	----	---

a	n	g	u	a	g	e	\0	p	r	e	s	e	n	t	e	d	\0	i	n
---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---



\0	e	l	e	t	r	o	n	i	c	\0	f	o	r	m	.	\0
----	---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----



delimiters: “ \”().”



start position



global pointer

A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a	\0	c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	----	---	---	---	---

e	c	t	i	o	n	\0	o	f	\0	t	e	x	t	s	\0	o	f	\0	w
---	---	---	---	---	---	----	---	---	----	---	---	---	---	---	----	---	---	----	---

r	e	t	t	e	n	\0	\0	o	r	\0	s	p	o	k	e	n	\0	\0	l
---	---	---	---	---	---	----	----	---	---	----	---	---	---	---	---	---	----	----	---

a	n	g	u	a	g	e	\0	p	r	e	s	e	n	t	e	d	\0	i	n
---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---



\0	e	l	e	t	r	o	n	i	c	\0	f	o	r	m	\0	\0
----	---	---	---	---	---	---	---	---	---	----	---	---	---	---	----	----



Return address of start position.

“form”

delimiters: “ \”().”



start position



global pointer

A	\0	\0	c	o	r	p	u	s	\0	\0	i	s	\0	a	\0	c	o	l	l
---	----	----	---	---	---	---	---	---	----	----	---	---	----	---	----	---	---	---	---

e	c	t	i	o	n	\0	o	f	\0	t	e	x	t	s	\0	o	f	\0	w
---	---	---	---	---	---	----	---	---	----	---	---	---	---	---	----	---	---	----	---

r	e	t	t	e	n	\0	\0	o	r	\0	s	p	o	k	e	n	\0	\0	l
---	---	---	---	---	---	----	----	---	---	----	---	---	---	---	---	---	----	----	---

a	n	g	u	a	g	e	\0	p	r	e	s	e	n	t	e	d	\0	i	n
---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---

\0	e	l	e	t	r	o	n	i	c	\0	f	o	r	m	\0	\0
----	---	---	---	---	---	---	---	---	---	----	---	---	---	---	----	----



Return address of start position.

NULL

- include <cstring> or <string.h>
- first time use **string variable** as parameter to setup global pointer
- others use **NULL** as parameter to avoid changing global pointer
- strtok will **modify the original string**
- **whitespace** character (\n, \r, \t ...)



# Practice Now

POJ 3981 - 字符串替换

Thank You for Your  
Listening.

