


The image features two large, thick, black L-shaped brackets. One is positioned on the left side, with its vertical bar extending downwards and its horizontal bar extending to the right. The other is on the right side, with its vertical bar extending upwards and its horizontal bar extending to the left. These brackets frame the central text.

Second tutorial after midterm
result release



Stand up and do something

That's appropriate

Tutorial 08

Xiao Liang | *yuxiao@comp.nus.edu.sg*

<https://github.com/yxliang01/cs1010-fun-stuff>

Adapted from Evan Tay's Slide

Today's plan

- Unit 20: C Pre-processor
 - *Problem Set 20.1, 20.2*
- Unit 21: Assert
 - *Problem Set 21.1*
- Unit 22: Efficiency
 - *Problem Set 22.1, 22.2*



UNIT 20

C PRE-PROCESSOR

Recap. PS 20.1. PS 20.2



Recap

- Preprocessor directive
 - *A directive which starts with #*
 - *To **#include** a file or,*
 - *To **#define** a macro*

Recap - #include

- #include <stdbool.h>
- #include "cs1010.h"

Recap - #define macro

■ #define NAME EXPRESSION

- *Use it to define macros which are repeatedly used in code.*
- *Whenever you use NAME in your code, it will be replaced with EXPRESSION*

Recap - #define macro

```
#define SQUARE(x) x*x  
#define PI 3.1415926  
  
int main() {  
    double radius = 4.0;  
    cs1010_print_double(PI*SQUARE(radius));  
}
```

Recap - #define macro

```
#define SQUARE(x) x*x
```

```
#define PI 3.1415926
```

```
int main() {
```

```
    double radius = 4.0;
```

```
    cs1010_print_double(PI*radius*radius);
```

```
}
```

Recap – Macro warnings

- Given:

```
#define SQUARE(x) x*x
```

- SQUARE(radius + 2) evaluates to:
radius + 2*radius + 2

Recap – Macro warnings

- Given:

```
#define SQUARE(x) ((x)*(x))
```

- SQUARE(radius + 2) evaluates to:

```
((radius + 2)*(radius + 2))
```

ENDING NOTE

ALWAYS USE
UPPERCASE WHEN
`#define` CONSTANTS
and MACROS



UNIT 20

C PRE-PROCESSOR

Recap. PS 20.1. PS 20.2



Problem Set 20.1 a)

```
#define MIN(a,b) a < b ? a : b
```

```
long i = MIN(10, 20);
```

```
long j = MIN(10, 20) + 1;
```

- What are the values of **i** and **j**?

Problem Set 20.1 a)

```
#define MIN(a,b) a < b ? a : b
```

```
long i = MIN(10, 20);
```

```
long j = MIN(10, 20) + 1;
```

■ What are the values of **i** and **j**?

– $i = 10 < 20 ? 10 : 20 = 10$

– $j = 10 < 20 ? 10 : 20 + 1$

$j = 10 < 20 ? 10 : 21 = 10$

Problem Set 20.1 a)

```
#define MIN(a,b) (a < b ? a : b)
```

```
long i = MIN(10, 20);
```

```
long j = MIN(10, 20) + 1;
```

■ What are the values of **i** and **j**?

– $i = 10 < 20 ? 10 : 20 = 10$

– $j = (10 < 20 ? 10 : 20) + 1$

$j = 10 + 1 = 11$

Problem Set 20.1 b)

```
#define MIN(a,b) a < b ? a : b
```

```
long i = 10;
```

```
long j = 20;
```

```
long k = MIN(j, i++);
```

■ What are the values of **i** and **k**?

a) $i = 11, k = 10$ b) $i = 11, k = 11$

c) $i = 12, k = 10$ d) $i = 12, k = 11$

Problem Set 20.1 b)

```
#define MIN(a,b) a < b ? a : b
```

```
long i = 10;
```

```
long j = 20;
```

```
long k = MIN(j, i++);
```

■ What are the values of **i** and **k**?

a) $i = 11, k = 10$ b) $i = 11, k = 11$

c) $i = 12, k = 10$ **d) $i = 12, k = 11$**

NANI?

Problem Set 20.1 b)

```
#define MIN(a,b) a < b ? a : b
```

```
long i = 10; long j = 20;
```

```
long k = MIN(j, i++);
```

- $k = 20 < i++ ? 20 : i++$
- $k = 20 < 10++ ? 20 : i++$
- $k = 20 < 10 ? 20 : 11++$
- $k = 11; i = 12$



UNIT 20

C PRE-PROCESSOR

Recap. PS 20.1. PS 20.2



Problem Set 20.2

■ Original solution

```
#define SWAP(T, x, y) {\n    T temp;\n    temp = x;\n    x = y;\n    y = temp;\n}\n\nint main() {\n    long x = 3.0; long y = -1.0;\n    SWAP(long, x, y);\n}
```

■ Modified version

```
#define SWAP(T, x, y) T temp = x;\n    x = y;\n    y = temp;\n\nint main() {\n    long x = 3.0; long y = -1.0;\n    SWAP(long, x, y);\n}
```

■ What could go wrong?

Problem Set 20.2 - Original

■ Original solution

```
#define SWAP(T, x, y) {\n    T temp;\n    temp = x;\n    x = y;\n    y = temp;\n}\n\nint main() {\n    long x = 3.0; long y = -1.0;\n    SWAP(long, x, y);\n}
```

■ Becomes:

```
int main() {\n    long x = 3.0; long y = -1.0;\n    {\n        long temp;\n        temp = x;\n        x = y;\n        y = temp;\n    }\n};\n}
```

Problem Set 20.2 - Original

■ Modified version

```
#define SWAP(T, x, y) T temp = x;\n    x = y;\n    y = temp;\n\nint main() {\n    long x = 3.0; long y = -1.0;\n    SWAP(long, x, y);\n}
```

■ Becomes:

```
int main() {\n    long x = 3.0; long y = -1.0;\n    long temp = x;\n    x = y;\n    y = temp;\n}
```

■ What could go wrong?

Problem Set 20.2 - Original

- Original becomes:

```
int main() {  
    long x = 3.0; long y = -1.0;  
    {  
        long temp;  
        temp = x;  
        x = y;  
        y = temp;  
    };  
}
```

- Second solution becomes:

```
int main() {  
    long x = 3.0; long y = -1.0;  
    long temp = x;  
    x = y;  
    y = temp;  
}
```

- What could go wrong?

Problem Set 20.2 - Original

- Original becomes:

```
int main() {  
    long temp = 5.0;  
    long x = 3.0; long y = -1.0;  
    {  
        long temp;  
        temp = x;  
        x = y;  
        y = temp;  
    };  
}
```

- Second solution becomes:

```
int main() {  
    long temp = 5.0;  
    long x = 3.0; long y = -1.0;  
    long temp = x;  
    x = y;  
    y = temp;  
}
```

- What could go wrong? This is what happens:
 - **error: redefinition of 'temp'**



UNIT 21

C PRE-PROCESSOR

Recap. PS 21.1.



Recap

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

```
int main(){
    char answer;
    printf("Is CS1010 hard? Enter Y/N: ");
    scanf("%c", answer);
    assert(answer == 'N');
}
```

- If wrong answer is given: **Assertion 'answer == 'Y'' failed.**



UNIT 21

C PRE-PROCESSOR

Recap. PS 21.1.



Problem Set 21.1

```
void foo(long x) {  
    if (x % 2 == 0) {  
        ...  
    } else {  
        assert(x % 2 == 1);  
    }  
}
```

- Would the assert in Line 5 above ever fail?

Problem Set 21.1

```
void foo(long x) {  
    if (x % 2 == 0) {  
        ...  
    } else {  
        assert(x % 2 == 1);  
    }  
}
```

- Would the assert in Line 5 above ever fail? **Yes**
 - *Consider $x = -1$*

Problem Set 21.2

Pop quiz

What does `cat` do?



Fun stuff



```
3. lol
lol
$ lolcat --help

Usage: lolcat [OPTION]... [FILE]...

Concatenate FILE(s), or standard input, to standard output.
With no FILE, or when FILE is -, read standard input.

    --spread, -p <f>:  Rainbow spread (default: 3.0)
    --freq, -F <f>:    Rainbow frequency (default: 0.1)
    --seed, -S <i>:     Rainbow seed, 0 = random (default: 0)
    --animate, -a:      Enable psychedelics
    --duration, -d <i>: Animation duration (default: 12)
    --speed, -s <f>:    Animation speed (default: 20.0)
    --force, -f:         Force color even when stdout is not a tty
    --version, -v:       Print version and exit
    --help, -h:         Show this message

Examples:
  lolcat f - g      Output f's contents, then standard input, then g's contents.
  lolcat            Copy standard input to standard output.
  fortune | lolcat  Display a rainbow cookie.

Report lolcat bugs to <http://www.github.org/busyloop/lolcat/issues>
lolcat home page: <http://www.github.org/busyloop/lolcat/>
Report lolcat translation bugs to <http://speaklolcat.com/>

$ fortune | cowsay | lolcat

-----
/ Your own qualities will help prevent \
\ your advancement in the world.      /
-----

      ^__^
      (oo)\_______
      (_____)\\\
              ||----w |
              ||

$
```

``sl | lolcat`` on PE

Installing lolcat

C implementation: <https://github.com/jaseg/lolcat>

Just ``make`` and you will get ``lolcat`` on PE!



UNIT 22

EFFICIENCY

Recap. PS 22.1. PS 22.2



Recap

- In CS1010, we will focus on the efficiency of your code in two senses:
 - *First, your code should not perform redundant work and it should not repeat itself unnecessarily.*
 - *Second, your algorithm should run within a given Big-O running time.*



UNIT 22

EFFICIENCY

PS 22.1. PS 22.2



Problem Set 22.1

- Order the following functions in increasing rate of growth:

1. $\text{Log}_{10} n$

2. $\ln n$

3. \sqrt{n}

4. n

5. $n \ln n$

6. n^2

7. n^4

8. 2^n

9. e^n

10. $n!$

How about when it is w.r.t Big-O?



Problem Set 22.1

$$O(\ln n) == O(\log_{10} n) \ll O(\sqrt{n}) \ll O(n) \ll O(n \ln n) \\ \ll O(n^2) \ll O(n^4) \ll O(2^n) \ll O(e^n) \ll O(n!)$$



UNIT 22

EFFICIENCY

Recap. PS 22.1. PS 22.2



Problem Set 22.2 a)

```
for (long i = 0; i < n; i += 1) {  
    for (long j = 0; j < n; j += 2) {  
        cs1010_println_long(i + j);  
    }  
}
```

- What is the Big-O running time of the following code, in terms of n ?

Problem Set 22.2 a)

```
for (long i = 0; i < n; i += 1) {  
    for (long j = 0; j < n; j += 2) {  
        cs1010_println_long(i + j);  
    }  
}
```

- What is the Big-O running time of the following code, in terms of n ? n^2

Problem Set 22.2 b)

```
for (long i = 0; i < n; i *= 2) {  
    for (long j = 0; j < n; j *= 2) {  
        cs1010_println_long(i + j);  
    }  
}
```

- What is the Big-O running time of the following code, in terms of n ?

Problem Set 22.2 b)

```
for (long i = 0; i < n; i *= 2) {  
    for (long j = 0; j < n; j *= 2) {  
        cs1010_println_long(i + j);  
    }  
}
```

- What is the Big-O running time of the following code, in terms of n ? $(\log_2 n)^2$

Problem Set 22.2 c)

```
long k = 1;
for (long j = 0; j < n; j += 1) {
    k *= 2;
    for (long i = 0; i < k; i += 1) {
        cs1010_println_long(i + j);
    }
}
```

- What is the Big-O running time of the following code, in terms of n ?

Problem Set 22.2 c)

```
long k = 1;
for (long j = 0; j < n; j += 1) {
    k *= 2;
    for (long i = 0; i < k; i += 1) {
        cs1010_println_long(i + j);
    }
}
```

- What is the Big-O running time of the following code, in terms of n ? 2^n

Text version solution

<https://github.com/yxliang01/cs1010-fun-stuff/blob/master/tutorial/08/Tut08.pdf>



Assignment 4



Guys, pls remember to free the memory... I wasted 7+ hrs just because someone didn't free the resource... And, that's y I'm still awake!!!

At 5:38AM

Still affecting my project(s) at the moment...

? question ☆

187 views

I hope everyone is having a good day today

mp4

edit

· good question | 11

Updated 1 day ago by Anonymous

S **the students' answer,** *where students collectively construct a single answer*

Segmentation Fault

edit

· undo thanks | 13

Updated 1 day ago by Anonymous



Vampire



Sunlight



Superman

@NPCCompleteTeens



Kryptonite



**software
engineers**

Writing proper
documentation

HAVE AN API



BUT NO DOCUMENTATION

memegenerator.net



Assignment 5





Q & A





THE END

<https://github.com/yxliang01/cs1010-fun-stuff/blob/master/tutorial-slides/tut08.pptx>

