C Programming
Day07.B

2017.09.26

for loop
functions

- for (S=0, i=1; i<11; ++i, S+=i) ;
  - not working
  - when S=0, S+=2 instead of S+=1
- for (S=0, i=1; i<11; S+=i, ++i) ;
  - the final i value is 11
  - S is incremented by 10, before incrementing i to 11
  - after `for` loop, S contains the correct value 55
- for (S=0, i=1; i<11; S+= i++) ;
  - combine two statements S+=i, ++i into one statement
  - S is incremented by old i, before incrementing i

S += i, i++

① S = S + i
② i = i + 1

S += i, ++i

① S = S + i.
② i = i + 1

S += i++

Operator Precedence (1)

S += i++

Precedence	Operator	Description	Associativity
1 highest	`::`	Scope resolution (C++ only)	None
2	`++`	Postfix increment	Left-to-right
	`--`	Postfix decrement	
	`()`	Function call	
	`[]`	Array subscripting	
	`.`	Element selection by reference	
	`->`	Element selection through pointer	
	`typeid()`	Run-time type information (C++ only) (see typeid)	
	`const_cast`	Type cast (C++ only) (see const_cast)	
	`dynamic_cast`	Type cast (C++ only) (see dynamic cast)	
	`reinterpret_cast`	Type cast (C++ only) (see reinterpret_cast)	
	`static_cast`	Type cast (C++ only) (see static_cast)	
3	`++`	Prefix increment	Right-to-left
	`--`	Prefix decrement	
	`+`	Unary plus	
	`-`	Unary minus	
	`!`	Logical NOT	
	`~`	Bitwise NOT (One's Complement)	
	`(type)`	Type cast	
	`*`	Indirection (dereference)	
	`&`	Address-of	
	`sizeof`	Size-of	
	`new`, `new[]`	Dynamic memory allocation (C++ only)	
	`delete`, `delete[]`	Dynamic memory deallocation (C++ only)	
4	`.*`	Pointer to member (C++ only)	Left-to-right
	`->*`	Pointer to member (C++ only)	
5	`*`	Multiplication	Left-to-right
	`/`	Division	
	`%`	Modulo (remainder)	
6	`+`	Addition	Left-to-right
	`-`	Subtraction	

Operator Precedence (2)

7	<<	Bitwise left shift	Left-to-right
	>>	Bitwise right shift	
8	<	Less than	Left-to-right
	<=	Less than or equal to	
	>	Greater than	
	>=	Greater than or equal to	
9	==	Equal to	Left-to-right
	!=	Not equal to	
10	&	Bitwise AND	Left-to-right
11	^	Bitwise XOR (exclusive or)	Left-to-right
12	|	Bitwise OR (inclusive or)	Left-to-right
13	&&	Logical AND	Left-to-right
14	||	Logical OR	Left-to-right
15	?:	Ternary conditional (see ?:)	Right-to-left
16	=	Direct assignment	Right-to-left
	+=	Assignment by sum	
	-=	Assignment by difference	
	*=	Assignment by product	
	/=	Assignment by quotient	
	%=	Assignment by remainder	
	<<=	Assignment by bitwise left shift	
	>>=	Assignment by bitwise right shift	
	&=	Assignment by bitwise AND	
	^=	Assignment by bitwise XOR	
	|=	Assignment by bitwise OR	
17	throw	Throw operator (exceptions throwing, C++ only)	Right-to-left
18 lowest	,	Comma	Left-to-right

*2

/2

s += i++

```
S += i++;
```

post prefix

```
S += i
```
① assign first

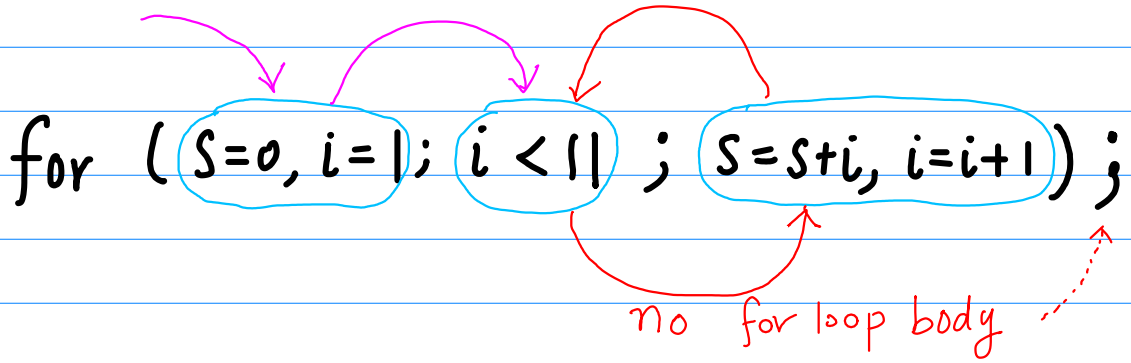
```
i++
```
② increment

① `S = s + i;`

② `i = i + 1;`

```
for (S=0, i=1; i<11 ; S=S+i, i=i+1);
```

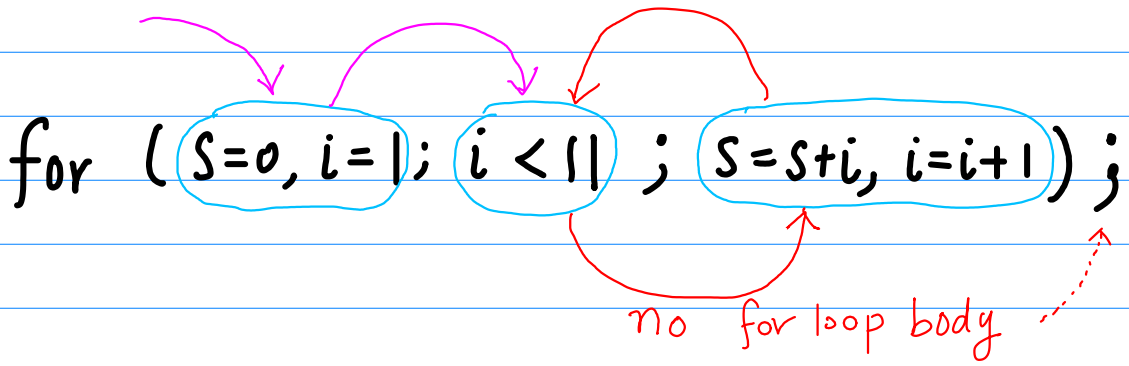
no for loop body

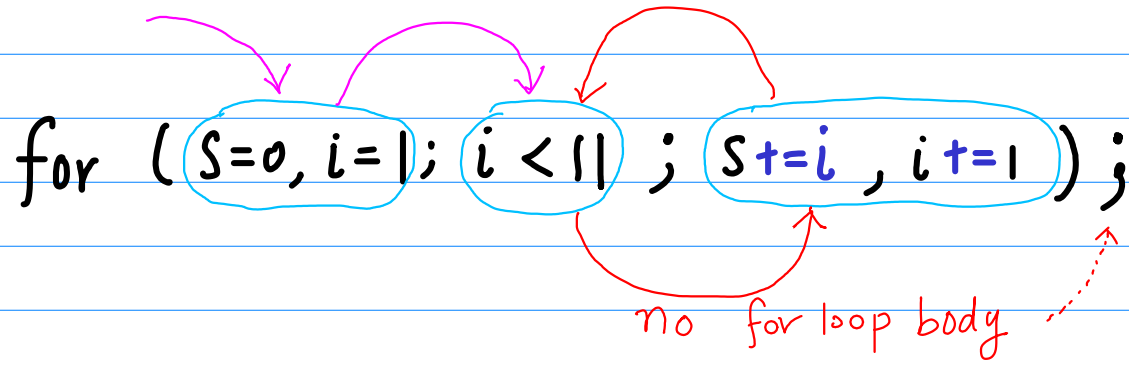


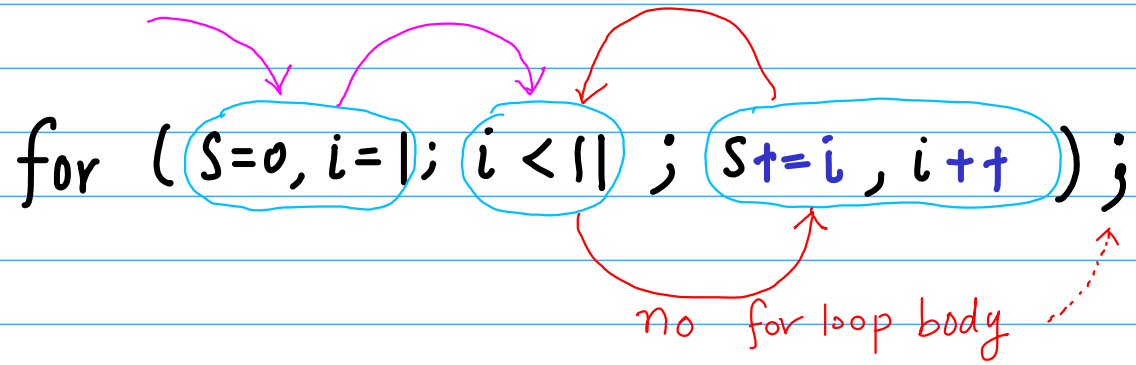
$S=0, i=1$	$i<11$	true	$S=S+1$	$i=1+1=2$
$S=1, i=2$	$i<11$	true	$S=S+2$	$i=2+1=3$
$S=3, i=3$	$i<11$	true	$S=S+3$	$i=3+1=4$
⋮	⋮	⋮	⋮	⋮
$S=45, i=10$	$i<11$	true	$S=S+10$	$i=10+1=11$
$S=55, i=11$	$i<11$	false		

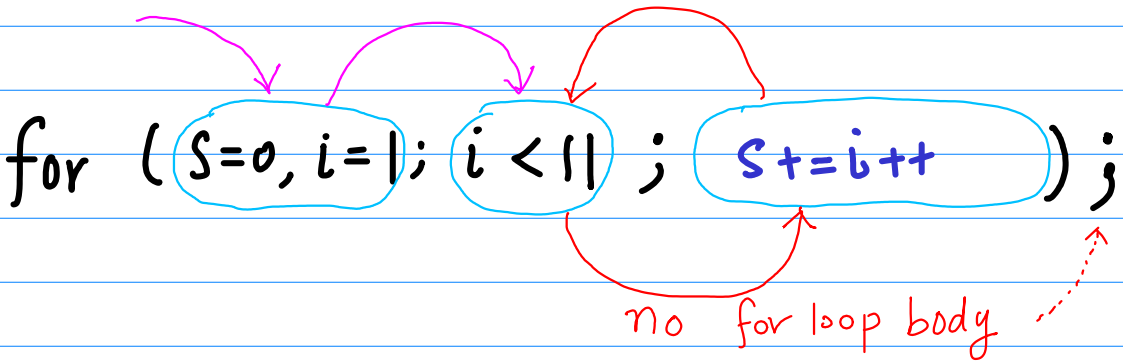
$i=1, \cdots, 10$

55


for (S=0, i=1; i<11 ; S=S+i, i=i+1) ;
no for loop body


for (S=0, i=1; i<11 ; S+=i , i+=1) ;
no for loop body


for (S=0, i=1; i<11 ; S+=i , i++) ;
no for loop body


for (S=0, i=1; i<11 ; S+=i++) ;
no for loop body

```c
#include <stdio.h>

int main(void) {
  int i=0, S=0;

  S=0;
  for (i=1; i<11; ++i) {
    S += i;
    printf("i=%d S=%d \n", i, S);
  }

  printf("----------------------\n");
  for (S=0, i=1; i<11; ++i, S+=i)   printf("i=%d S=%d (X) \n", i, S);
  printf("i=%d S=%d \n", i, S);

  printf("----------------------\n");
  for (S=0, i=1; i<11; S+=i, ++i)   printf("i=%d S=%d \n", i, S);
  printf("i=%d S=%d \n", i, S);

  printf("----------------------\n");
  for (S=0, i=1; i<11; S+=i++)      printf("i=%d S=%d \n", i, S);
  printf("i=%d S=%d \n", i, S);

  printf("----------------------\n");
}
```

i=1 S=1
i=2 S=3
i=3 S=6
i=4 S=10
i=5 S=15
i=6 S=21
i=7 S=28
i=8 S=36
i=9 S=45
i=10 S=55
---
i=1 S=0 (X)
i=2 S=2 (X)
i=3 S=5 (X)
i=4 S=9 (X)
i=5 S=14 (X)
i=6 S=20 (X)
i=7 S=27 (X)
i=8 S=35 (X)
i=9 S=44 (X)
i=10 S=54 (X)
---
i=1 S=0
i=2 S=1
i=3 S=3
i=4 S=6
i=5 S=10
i=6 S=15
i=7 S=21
i=8 S=28
i=9 S=36
i=10 S=45
---
i=1 S=0
i=2 S=1
i=3 S=3
i=4 S=6
i=5 S=10
i=6 S=15
i=7 S=21
i=8 S=28
i=9 S=36
i=10 S=45
---

looks like that it doesn't work

```c
#include <stdio.h>

int main(void) {
  int i=0, S=0;

  S=0;
  for (i=1; i<11; ++i) {
    S += i;
    printf("i=%d S=%d \n", i, S);
  }

  printf("----------------------\n");
  for (S=0, i=1; i<11; ++i, S+=i);  printf("i=%d S=%d (X) \n", i, S);

  printf("----------------------\n");
  for (S=0, i=1; i<11; S+=i, ++i);  printf("i=%d S=%d \n", i, S);

  printf("----------------------\n");
  for (S=0, i=1; i<11; S+=i++);  printf("i=%d S=%d \n", i, S);

  printf("----------------------\n");
}
```

```
i=1 S=1
i=2 S=3
i=3 S=6
i=4 S=10
i=5 S=15
i=6 S=21
i=7 S=28
i=8 S=36
i=9 S=45
i=10 S=55
----------------------
i=11 S=65 (X)
----------------------
i=11 S=55
----------------------
i=11 S=55
----------------------
```