

# Iteration (1A)

---

Copyright (c) 2010, 2011 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".

Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice.

# Sigma Notation and Flow Chart

$$S_n = \sum_{k=1}^n a_k$$
$$= a_1 + a_2 + a_3 + \dots + a_n$$

$$S \leftarrow 0 + a_1$$

$$S \leftarrow (0 + a_1) + a_2$$

$$S \leftarrow (0 + a_1 + a_2) + a_3$$

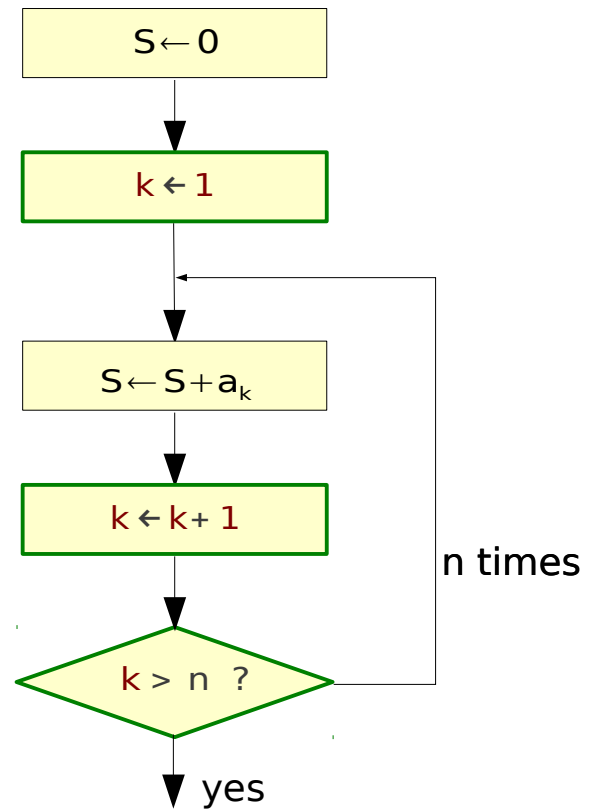
$$S \leftarrow (0 + a_1 + a_2 + a_3) + a_4$$

$$S \leftarrow (0 + a_1 + a_2 + a_3 + a_4) + a_5$$

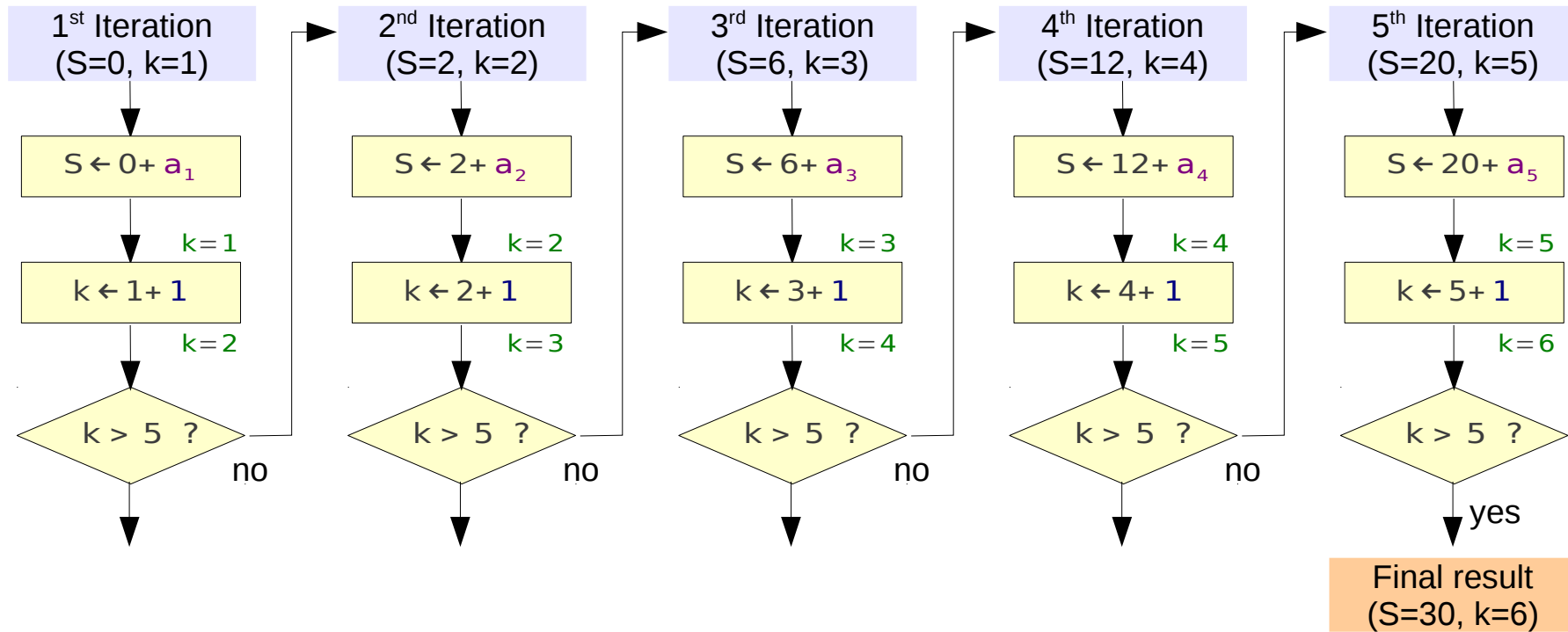
loop index  
initialization

loop index  
update

loop index  
condition check

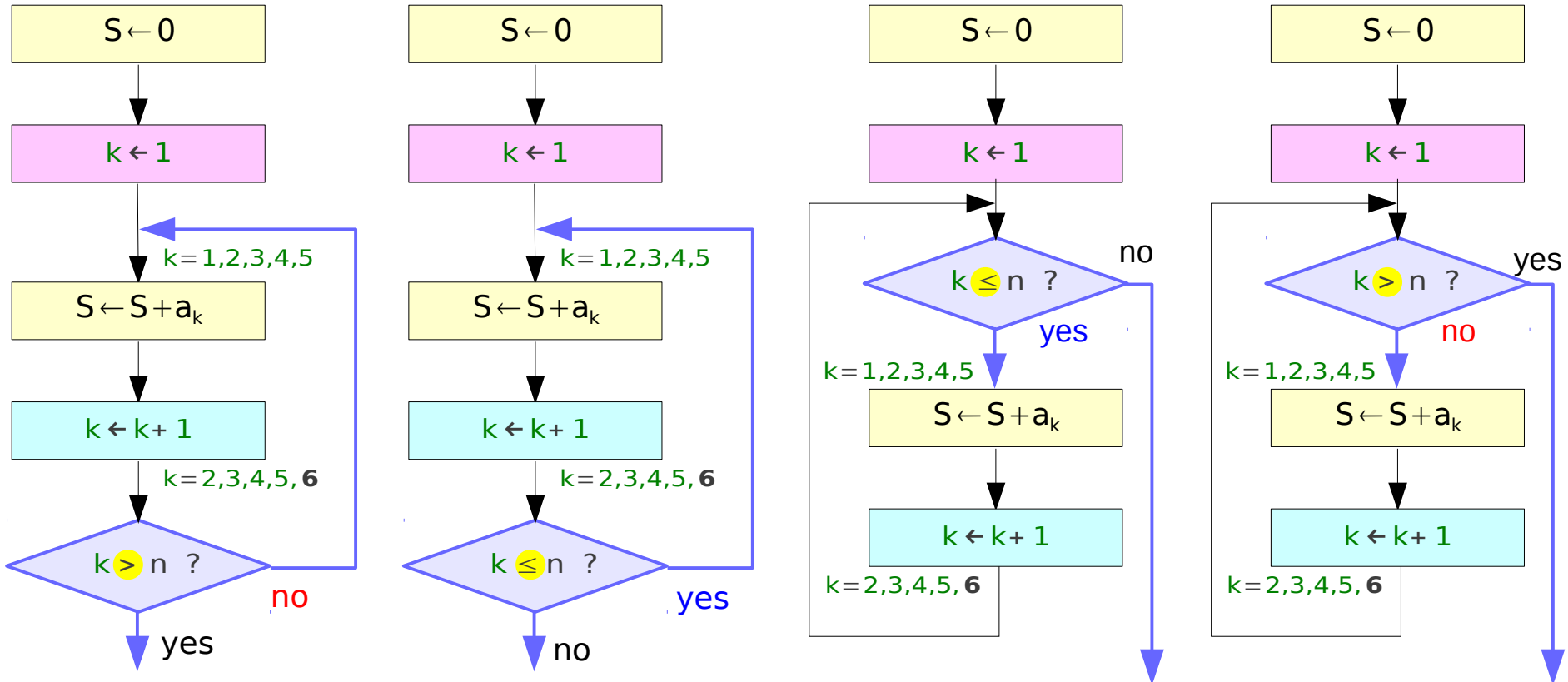


# Loop Unrolling

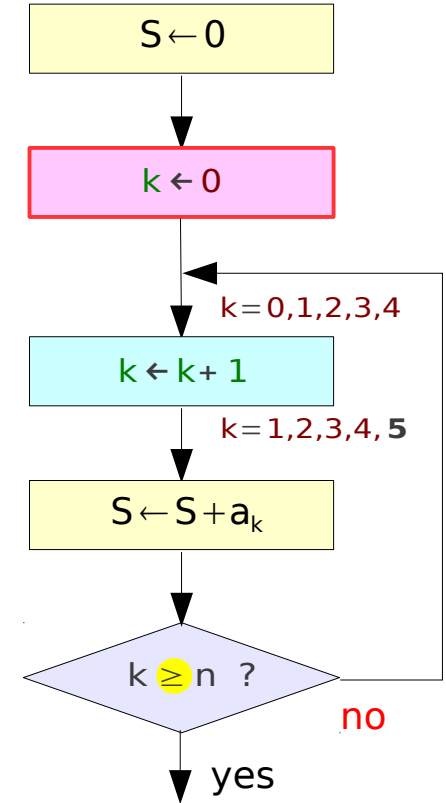
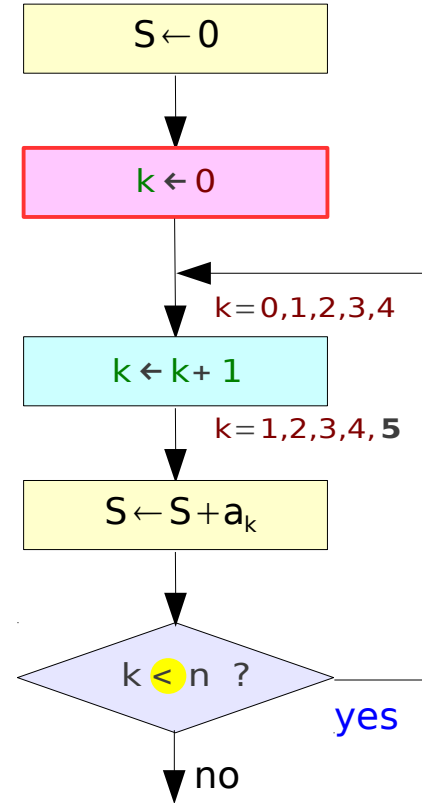
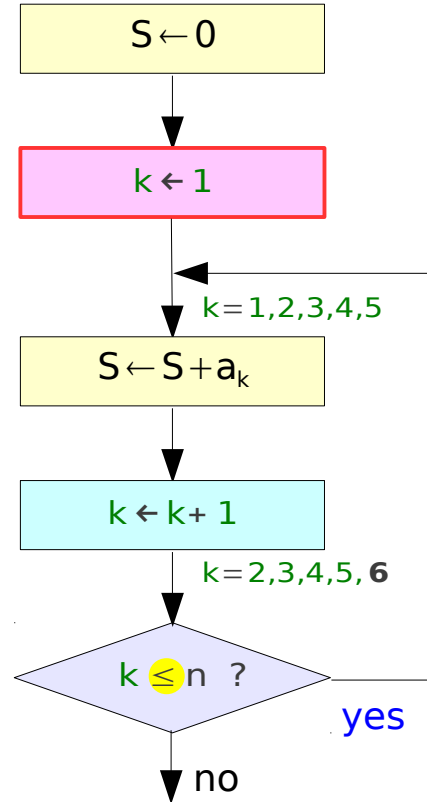
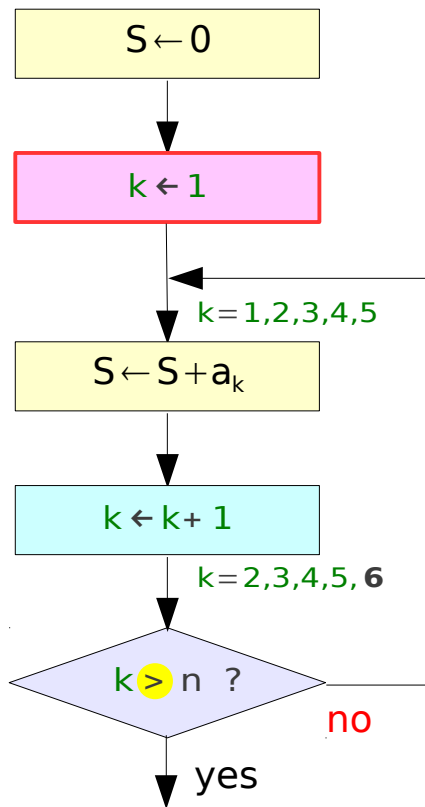


$a_1=2,$   
 $a_2=4,$   
 $a_3=6,$   
 $a_4=8,$   
 $a_5=10$

# Check Conditions

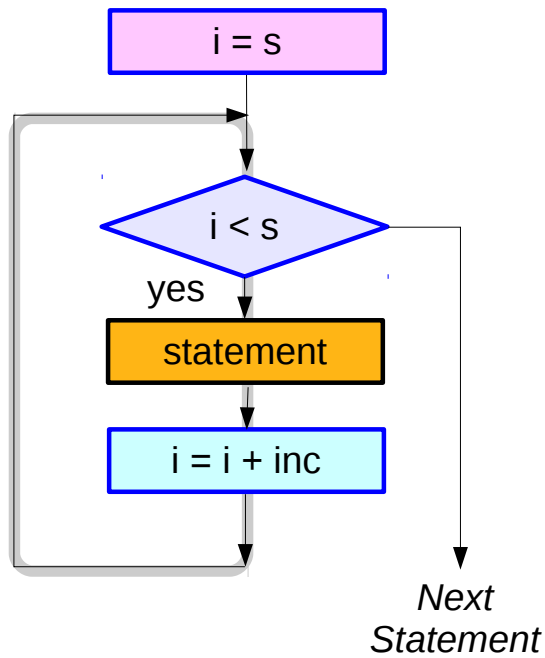


# Initial Conditions

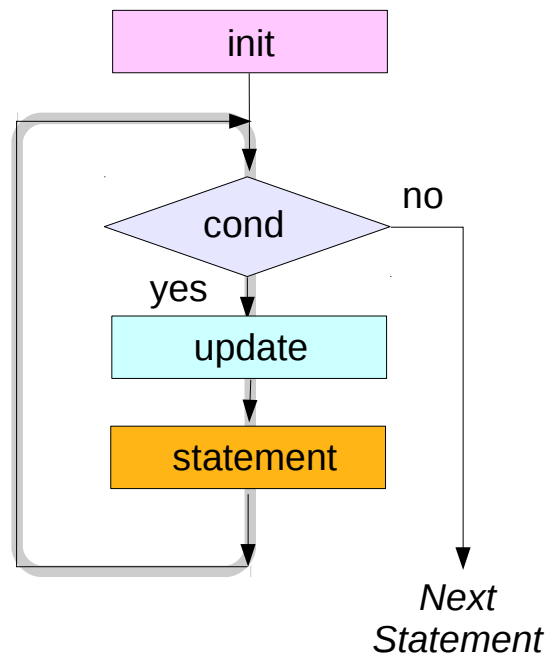


# Loop Statements

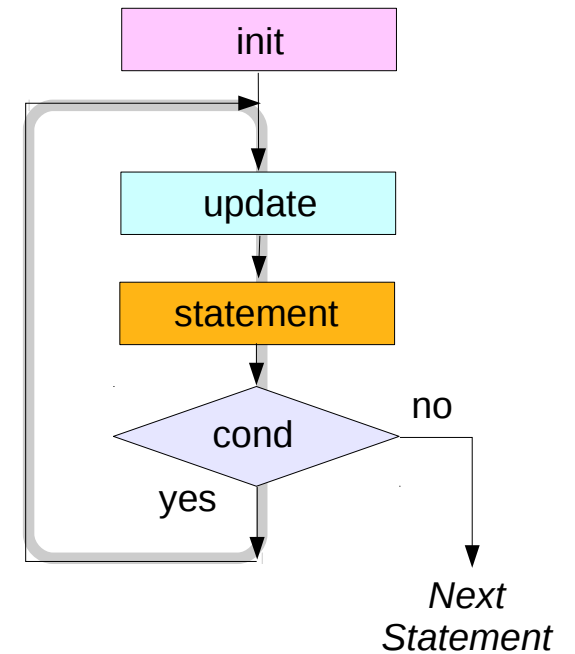
```
for i = s : inc : e  
  statements  
endfor
```



```
init  
while (cond)  
  update  
  statements  
endwhile
```



```
init  
do  
  update  
  statements  
until (cond)
```



# for loop

```
S = 0;  
for k = 0:4  
    S = S + k+1;  
endfor
```

```
S = 0;  
for k = 0:4  
    S = S + k;  
endfor
```



# for loop's expressions

```
for k = [ 1, 3, 2, 4]  
k;  
endfor
```

```
k = 1  
k = 3  
k = 2  
k = 4
```

```
for k = [ 1, 3; 2, 4]  
k;  
endfor
```

```
k = [1; 2]  
k = [3, 4]
```

## Range Expression

1 : 1 : 4

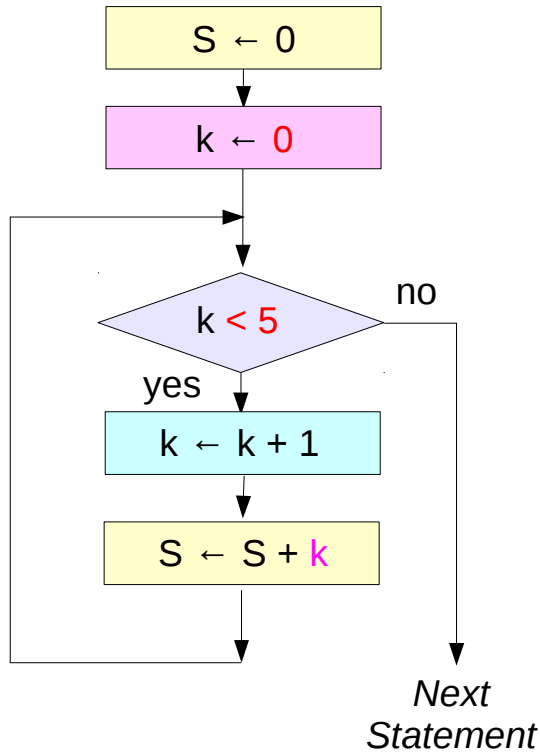
⇒ [ 1, 2, 3, 4]

4 : -1 : 1

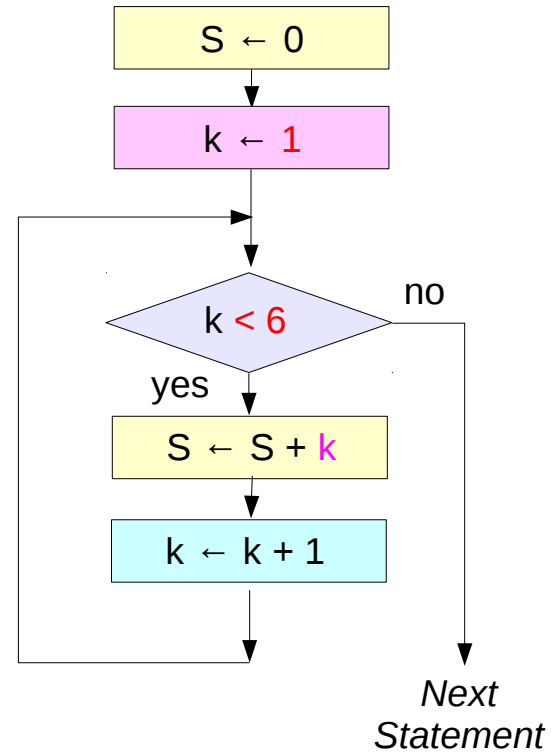
⇒ [ 4, 3, 2, 1]

# while loop

```
S = 0; k=0;  
while (k<5)  
    k = k+1;  
    S = S+k;  
endwhile
```



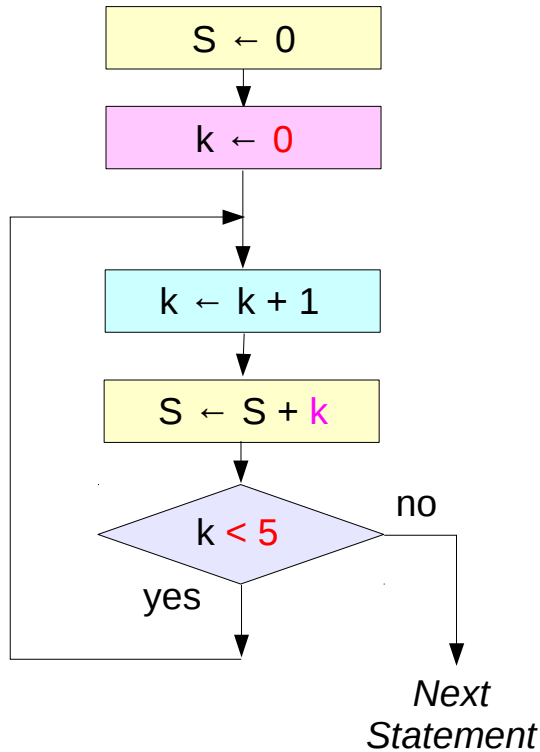
```
S = 0; k=1;  
while (k<6)  
    S = S+k;  
    k = k+1;  
endwhile
```



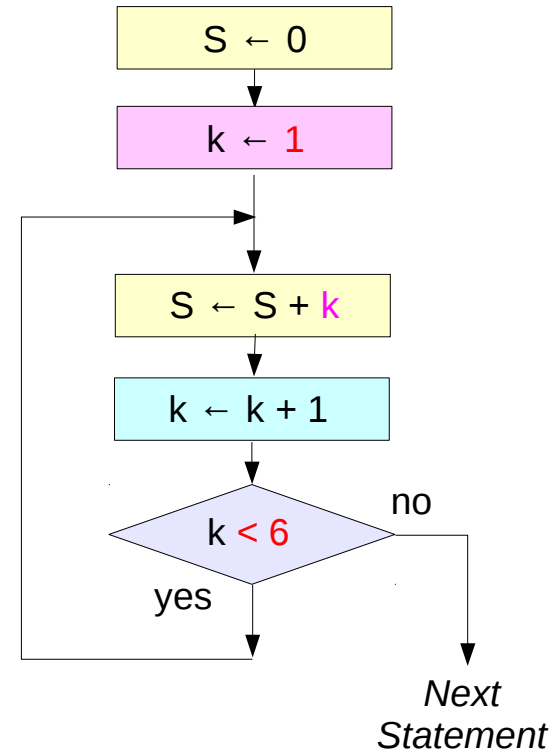
flexible

# do-while loop

```
S = 0; k=0;  
do  
    k = k+1;  
    S = S+k;  
until (k<5)
```



```
S = 0; k=1;  
do {  
    S = S+k;  
    k = k+1;  
until (k<6)
```



*flexible,  
at least  
once*

# Nested For Loop Examples

```
for i = 0 : 2
    printf("St1 \n"); endfor
for j = 0 : 3
    printf("St2 \n"); endfor
printf("St3 \n");
```

```
for i = 0 : 2
    printf("St1 \n");
    for j = 0 : 3
        printf("St2 \n"); endfor endfor
    printf("St3 \n");
```

```
for i = 0 : 2
    printf("St1 \n"); endfor
for j = 0 : 3
    printf("St2 \n");
    printf("St3 \n"); endfor
```

```
for i = 0 : 2
    printf("St1 \n");
endfor
for j = 0 : 4
    printf("St2 \n");
endfor
printf("St3 \n");
```

```
for i = 0 : 2
    printf("St1 \n");
    for j = 0 : 3
        printf("St2 \n");
    endfor
endfor
printf("St3 \n");
```

```
for i = 0 : 2
    printf("St1 \n");
endfor
for j = 0 : 3
    printf("St2 \n");
    printf("St3 \n");
endfor
```

## References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun