

# 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

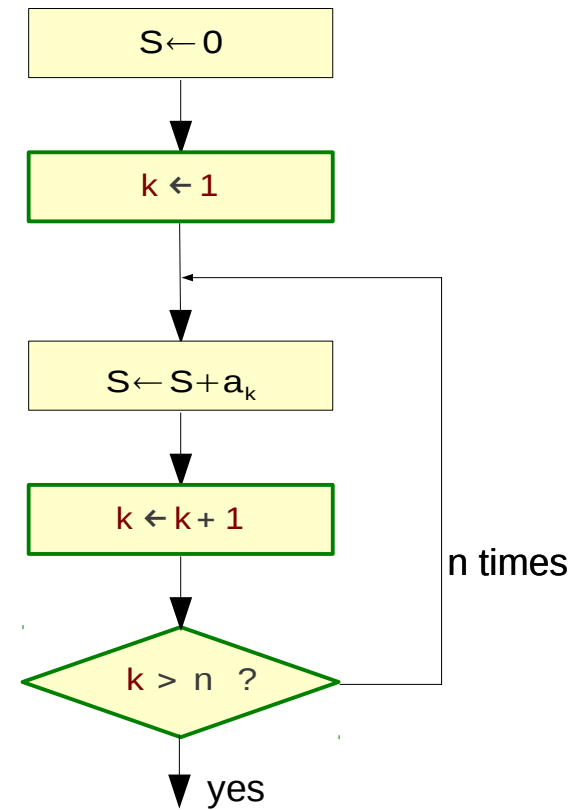
$$S_n = \sum_{k=1}^n a_k$$
$$= a_1 + a_2 + a_3 + \dots + a_n$$

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

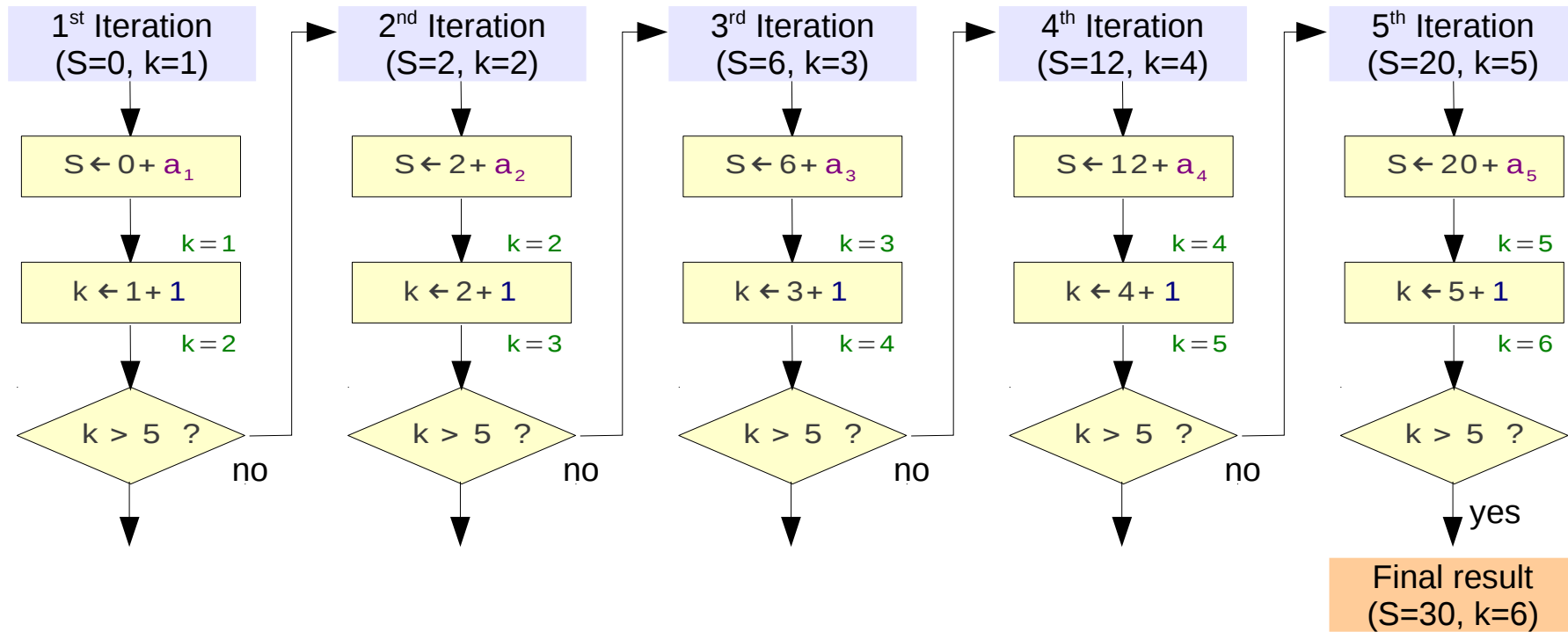
loop index  
initialization

loop index  
update

loop index  
condition check

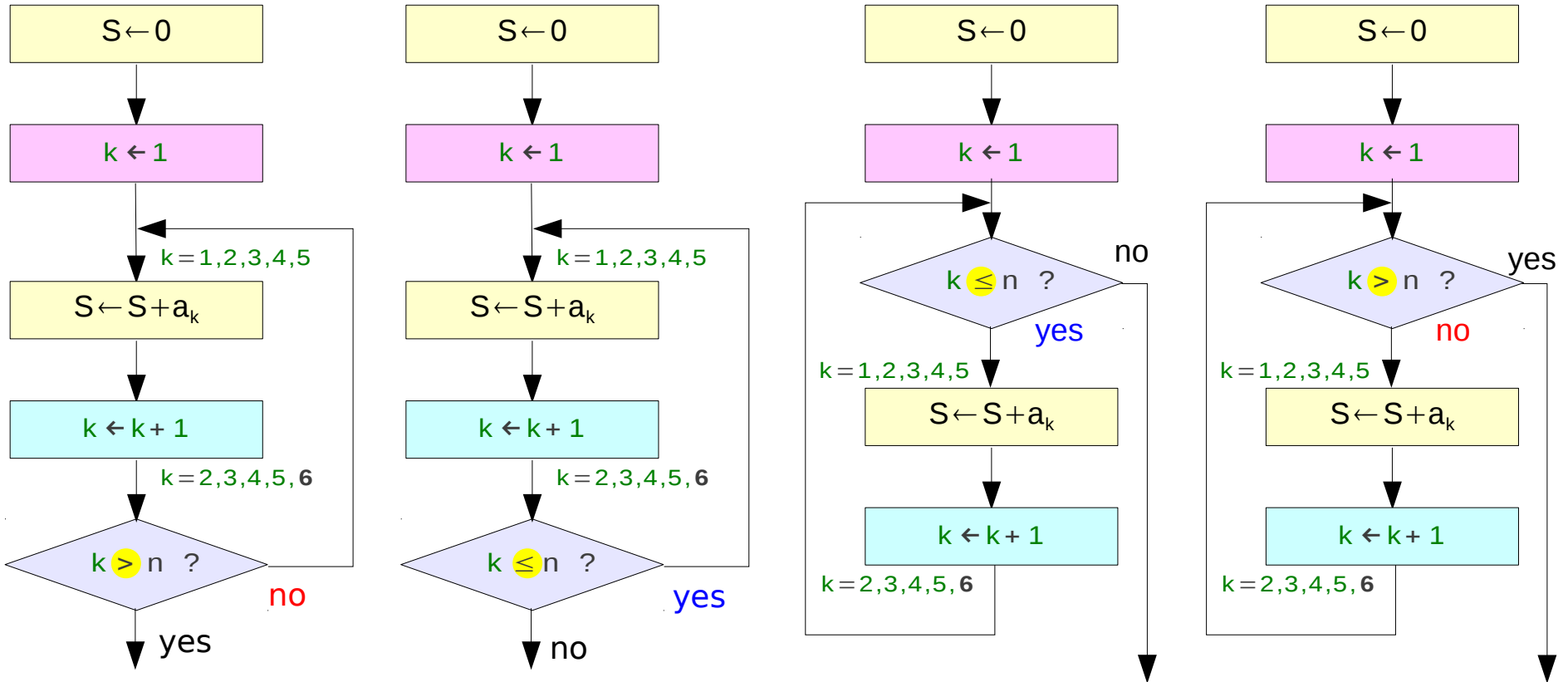


# Loop Unrolling

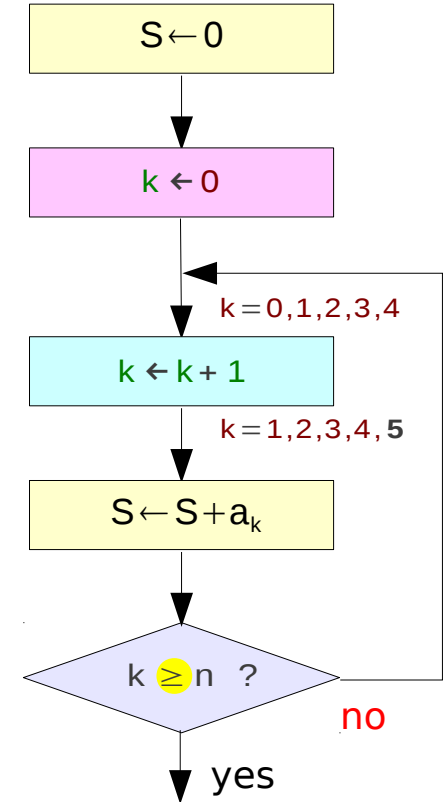
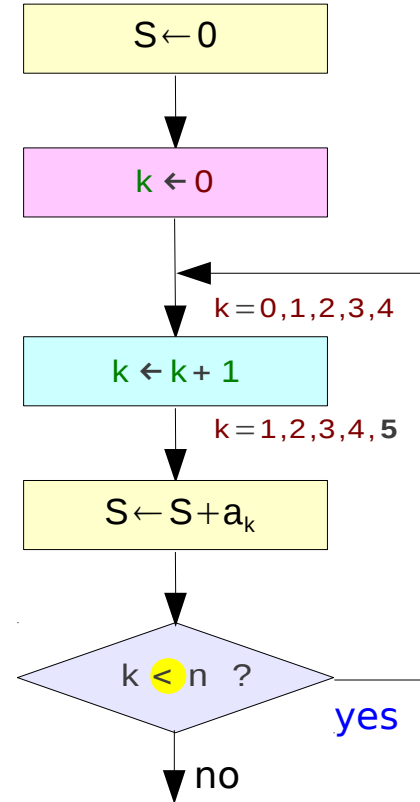
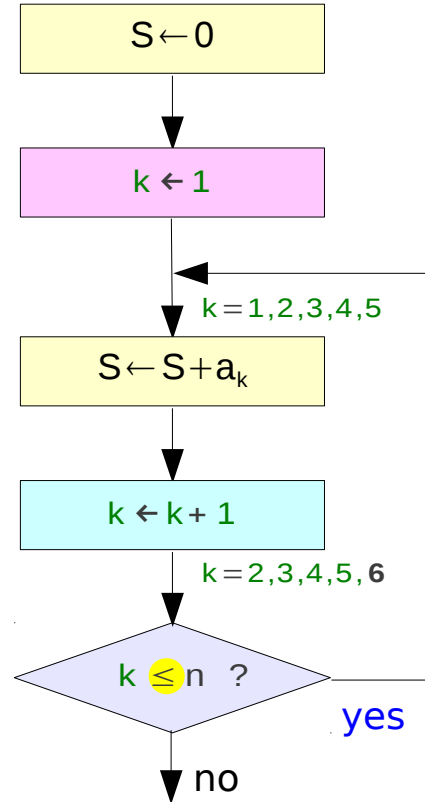
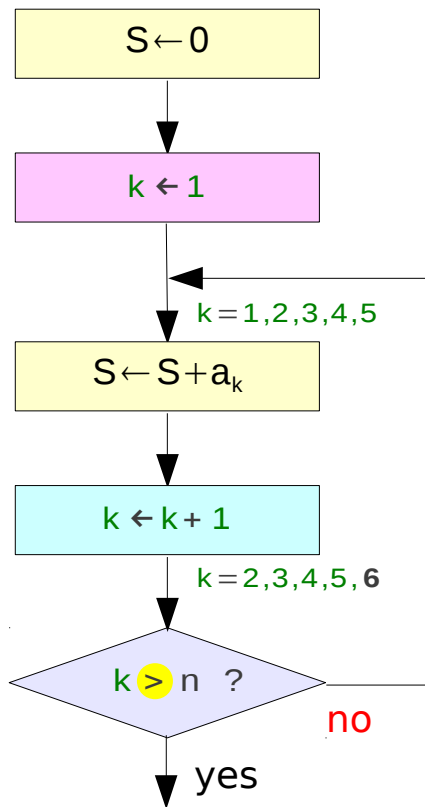


$a_1=2,$   
 $a_2=4,$   
 $a_3=6,$   
 $a_4=8,$   
 $a_5=10$

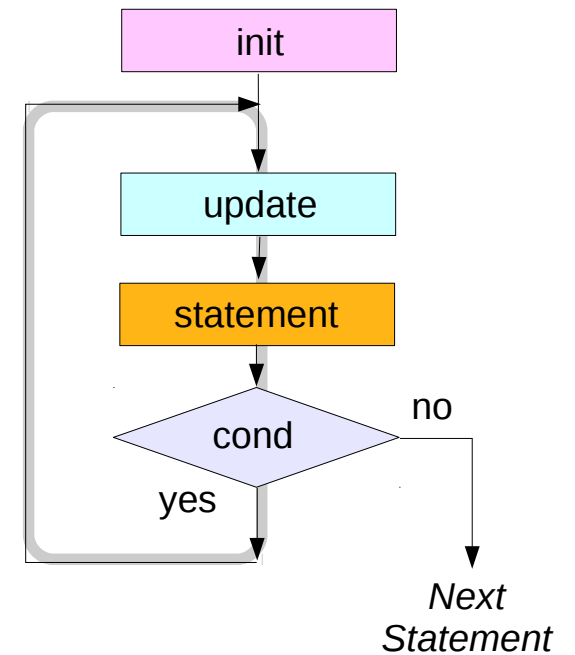
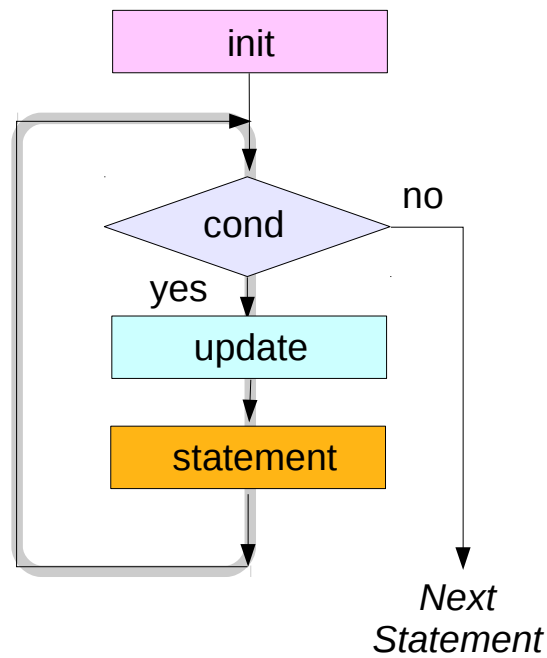
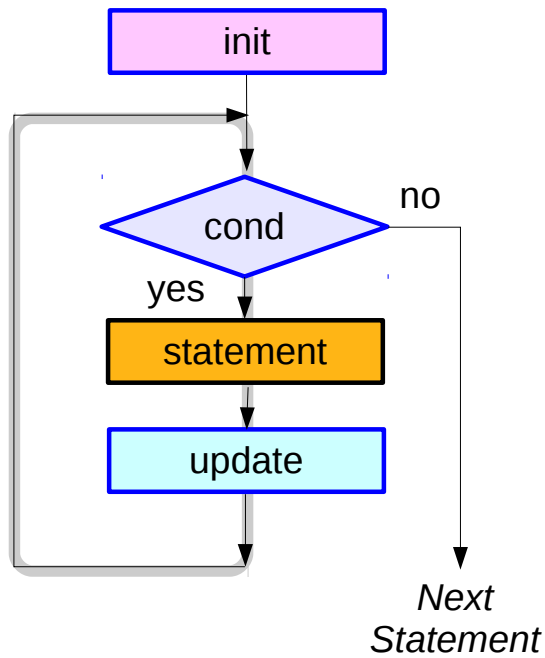
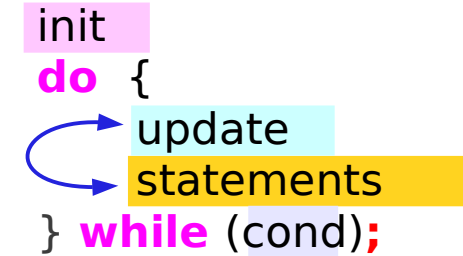
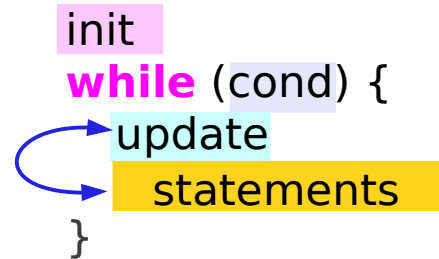
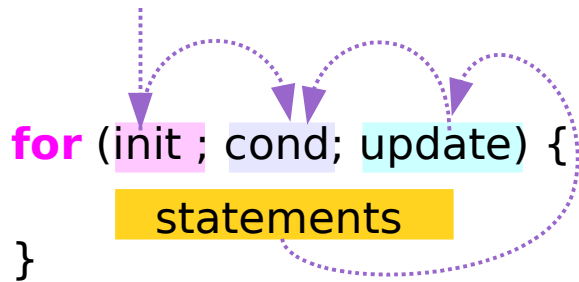
# Check Condition



# Initial Condition

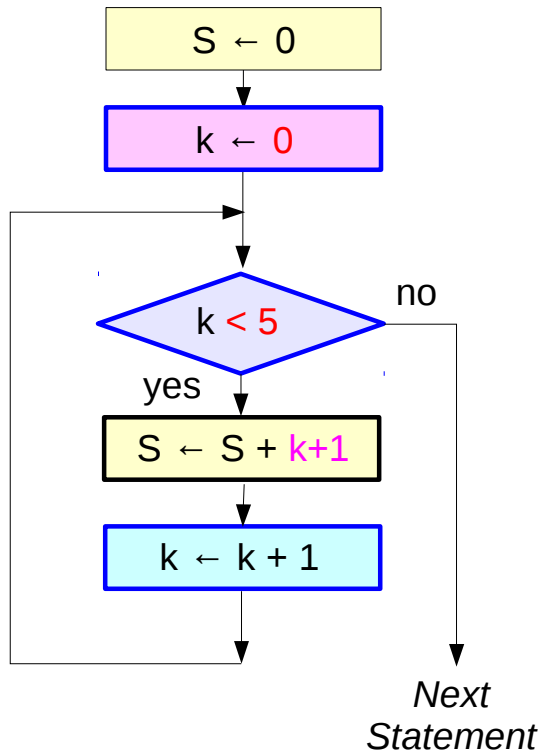


# Loop Statements

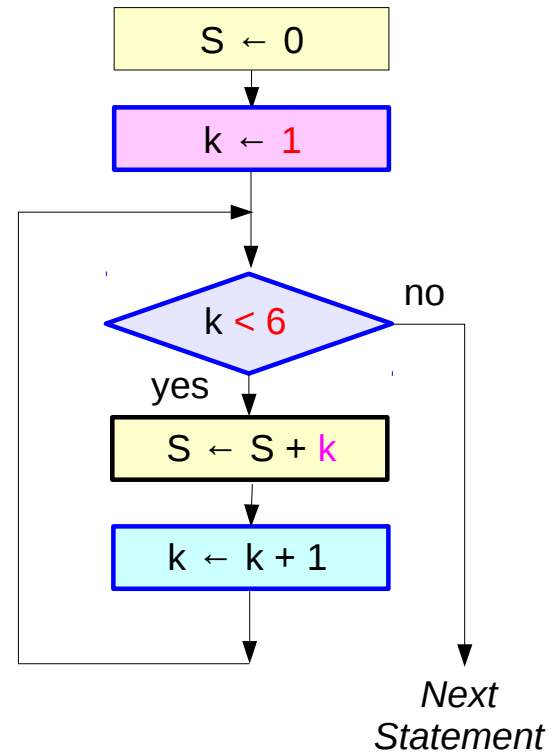


# for loop

```
S = 0;  
for (k=0; k<5; k++) {  
    S = S + k+1;  
}
```



```
S = 0;  
for (k=1; k<6; k++) {  
    S = S + k;  
}
```

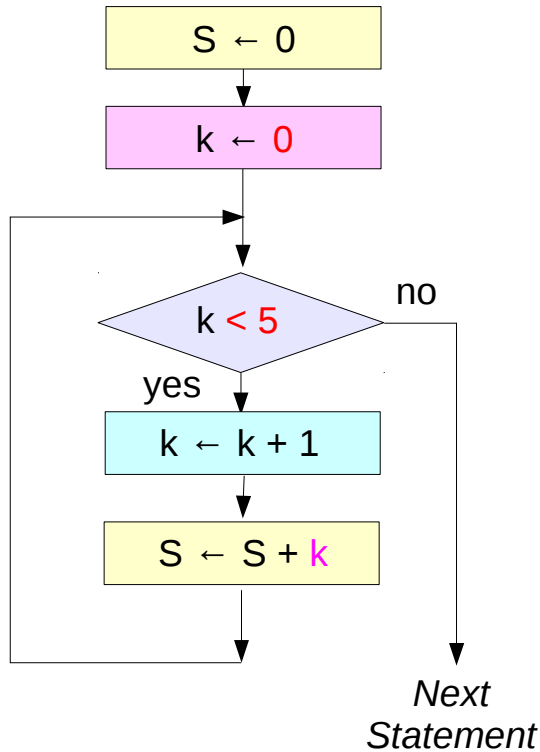


fixed

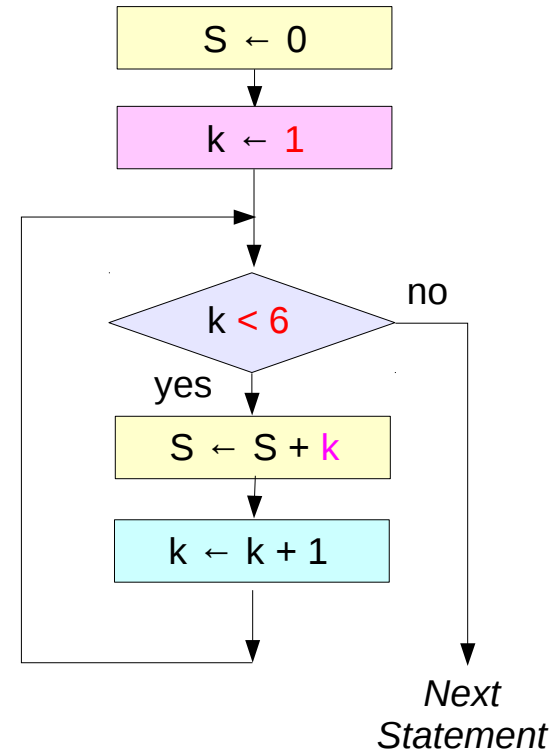


# while loop

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



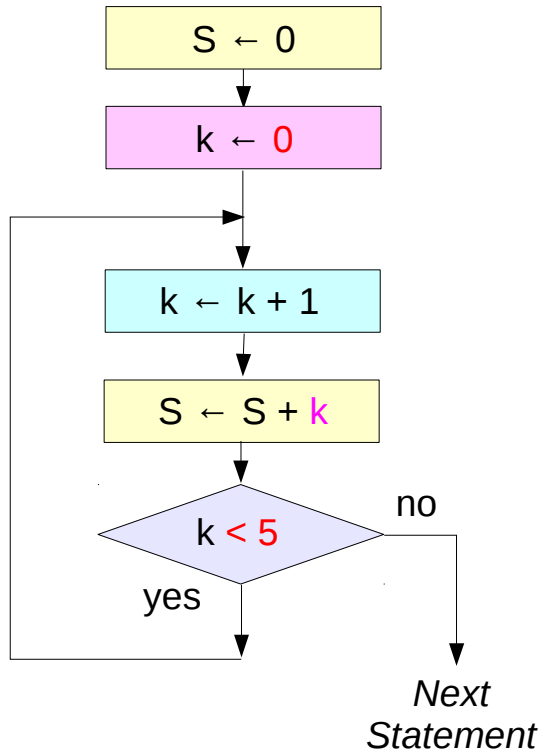
```
S = 0; k=1;  
while (k<6) {  
    S = S+k;  
    k = k+1;  
}
```



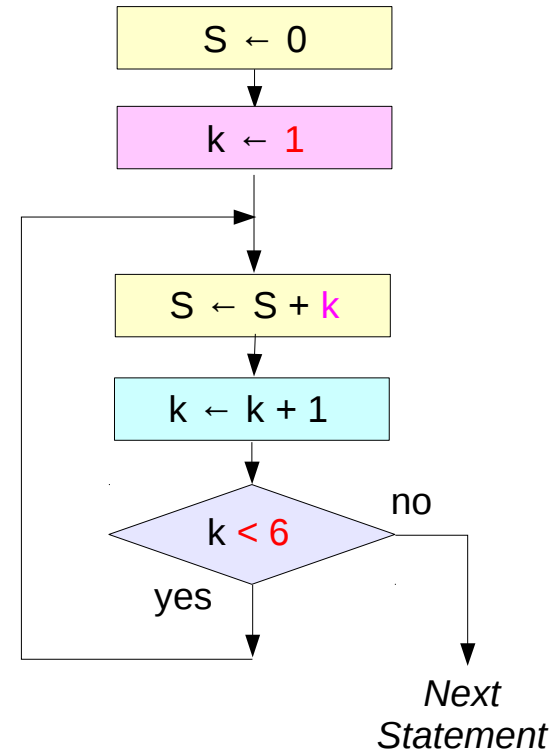
flexible

# do-while loop

```
S = 0; k=0;  
do {  
    k = k+1;  
    S = S+k;  
} while (k<5);
```



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



flexible,  
at least  
once

# Nested For Loop Examples

```
for (i=0; i<3; ++i)
printf("St1 \n");
for (j=0; j<4; ++j)
printf("St2 \n");
printf("St3 \n");
```

```
for (i=0; i<3; ++i) {
printf("St1 \n");
for (j=0; j<4; ++j)
printf("St2 \n"); }
printf("St3 \n");
```

```
for (i=0; i<3; ++i)
printf("St1 \n");
for (j=0; j<4; ++j) {
printf("St2 \n");
printf("St3 \n"); }
```

```
for (i=0; i<3; ++i)
    printf("St1 \n");
for (j=0; j<4; ++j)
    printf("St2 \n");
printf("St3 \n");
```

```
for (i=0; i<3; ++i) {
    printf("St1 \n");
    for (j=0; j<4; ++j)
        printf("St2 \n");
}
printf("St3 \n");
```

```
for (i=0; i<3; ++i)
    printf("St1 \n");
for (j=0; j<4; ++j) {
    printf("St2 \n");
    printf("St3 \n");
}
```

## 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