Iteration (1A)

Young Won Lim
9/25/17

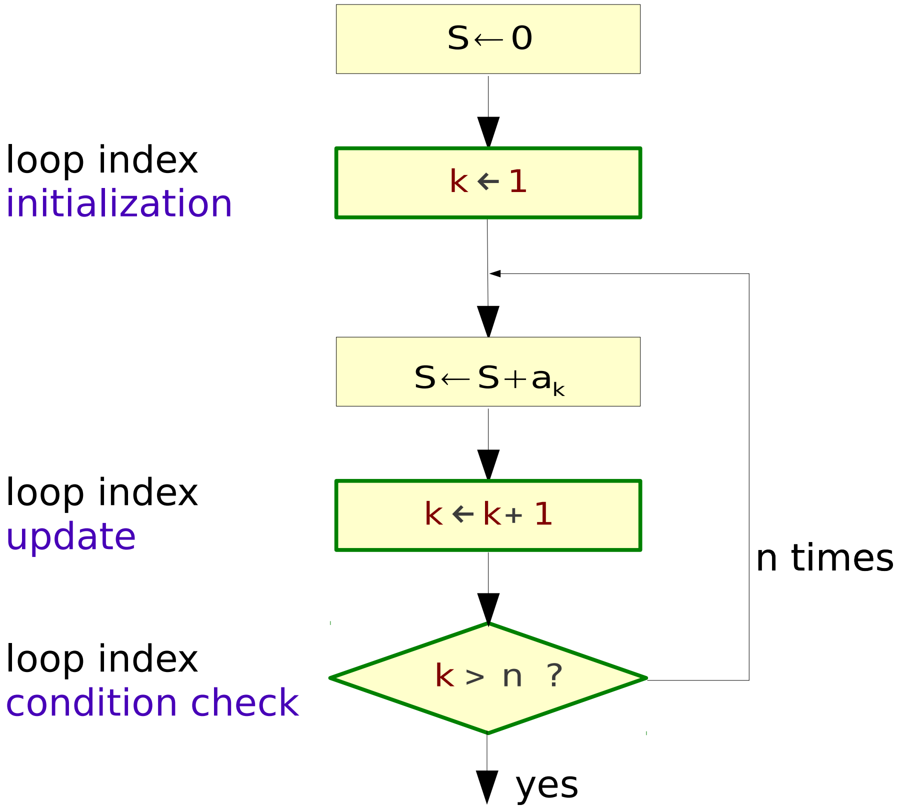
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Sigma Notation and Flow Chart

$$S_n = \sum_{k=1}^{n} a_k$$

$$= a_1 + a_2 + a_3 + \quad \cdots \quad + a_n$$



$S \leftarrow 0$

loop index
initialization

$k \leftarrow 1$

$S \leftarrow S + a_k$

loop index
update

$k \leftarrow k + 1$

loop index
condition check

$k > n$?

n times

yes

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

Adding one value at a time

$$\boxed{S = 0}$$

$$\boxed{S = S + a_1} \qquad ((((\underbrace{(a_1)}_{} + a_2) + a_3) + a_4) + a_5)$$

$$\boxed{S = S + a_2} \qquad (((\underbrace{((a_1) + a_2)}_{} + a_3) + a_4) + a_5)$$

$$\boxed{S = S + a_3} \qquad ((\underbrace{(((a_1) + a_2) + a_3)}_{} + a_4) + a_5)$$

$$\boxed{S = S + a_4} \qquad (\underbrace{((((a_1) + a_2) + a_3) + a_4)}_{} + a_5)$$

$$\boxed{S = S + a_5} \qquad \underbrace{(((((a_1) + a_2) + a_3) + a_4) + a_5)}_{}$$

$$(((((a_1) + a_2) + a_3) + a_4) + a_5)$$

Adding one value at a time

$$S = a_1 + a_2 + a_3 + a_4 + a_5$$

$$S = 0$$

$$S = S + a_1$$

$$S = S + a_2$$

$$S = S + a_3$$

$$S = S + a_4$$

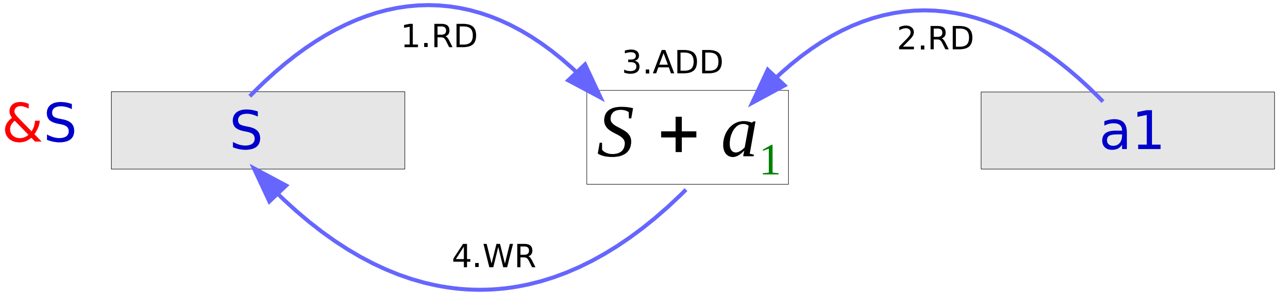
$$S = S + a_5$$

$$S = \left(\left(\left(\left(\left(a_1\right) + a_2\right) + a_3\right) + a_4\right) + a_5\right)$$

S = S + a1 statement

$$S = a_1 + a_2 + a_3 + a_4 + a_5$$

$$S = S + a_1$$



1. **Read** the current value of S

2. **Read** the current value of a1

3. **Add** the two values

4. **Write** the updated value of S

Accumulation Variable S

$$S = a_1 + a_2 + a_3 + a_4 + a_5$$

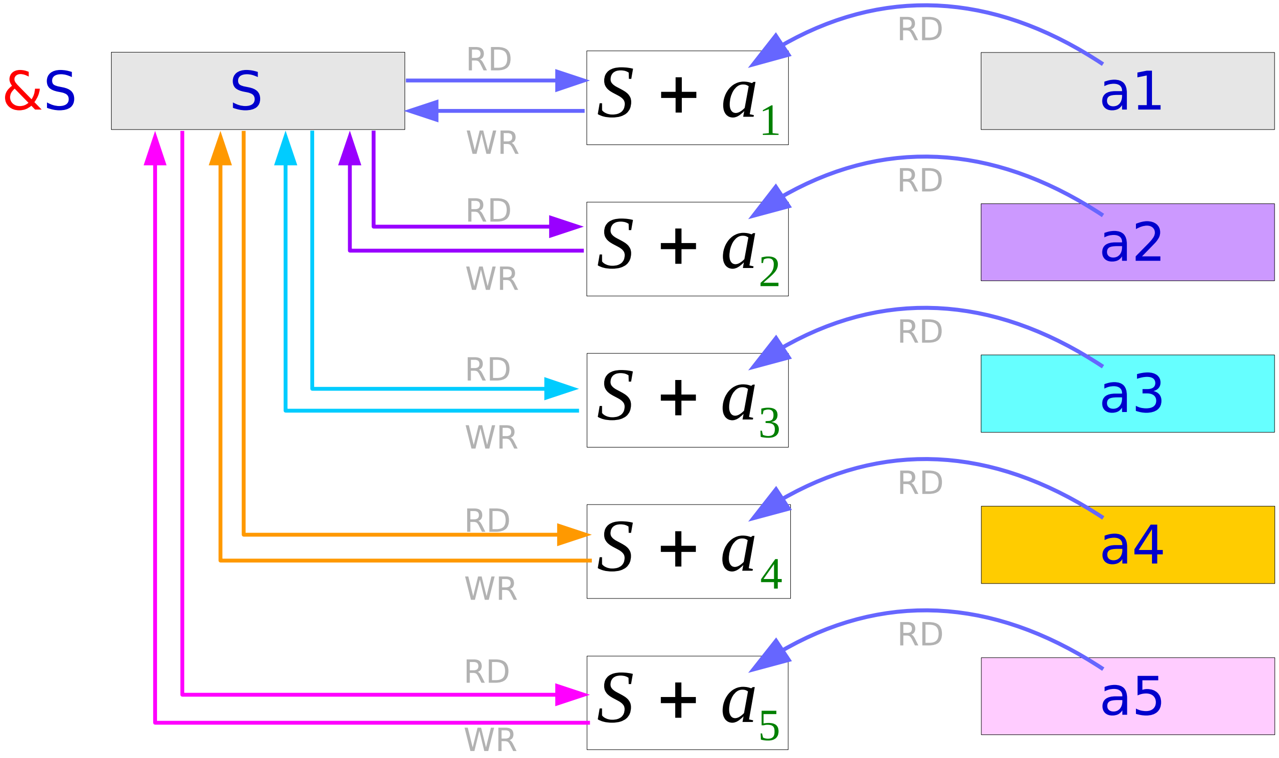
$$S = S + a_1$$

$$S = S + a_2$$

$$S = S + a_3$$

$$S = S + a_4$$

$$S = S + a_5$$



Using a counter variable k

$$S = a_1 + a_2 + a_3 + a_4 + a_5$$

$$\begin{array}{ll}
\boxed{S = 0} & \boxed{S = 0, \qquad k = 1} \\
S = S + a_1 & \boxed{S = S + a_k, \quad k = k + 1} \\
S = S + a_2 & \boxed{S = S + a_k, \quad k = k + 1} \\
S = S + a_3 & \boxed{S = S + a_k, \quad k = k + 1} \\
S = S + a_4 & \boxed{S = S + a_k, \quad k = k + 1} \\
S = S + a_5 & \boxed{S = S + a_k, \quad k = k + 1}
\end{array}$$

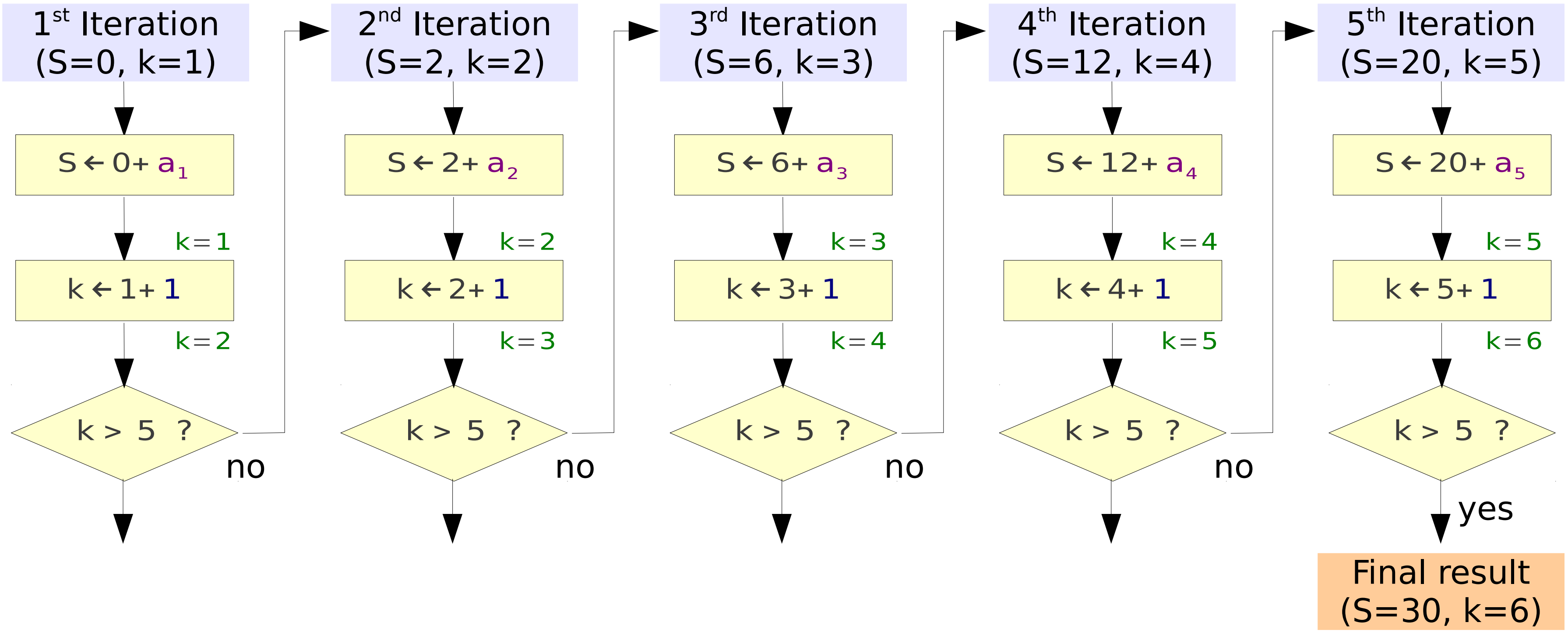
the same 5 statements are repeated

Repetition

Loop Statements

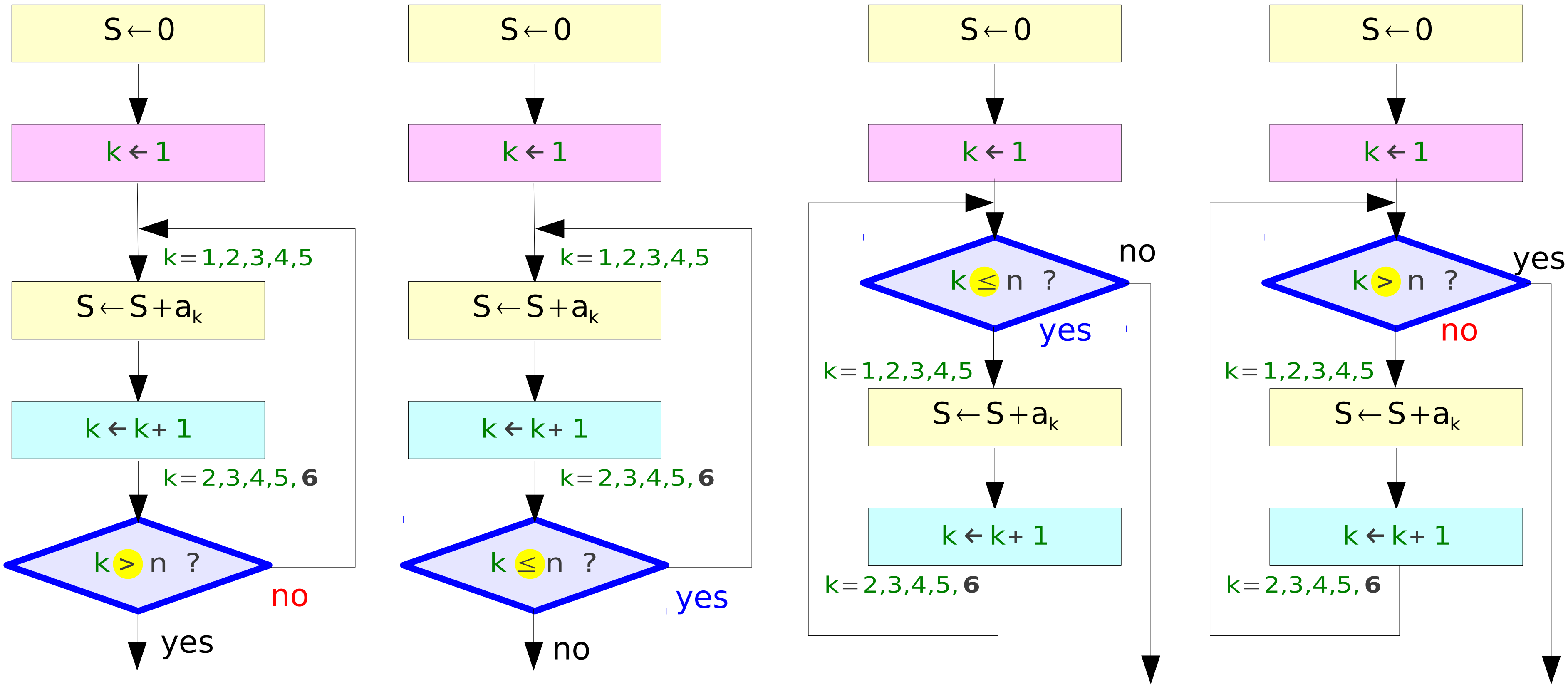
$$S = (((((a_1) + a_2) + a_3) + a_4) + a_5)$$

Loop Unrolling

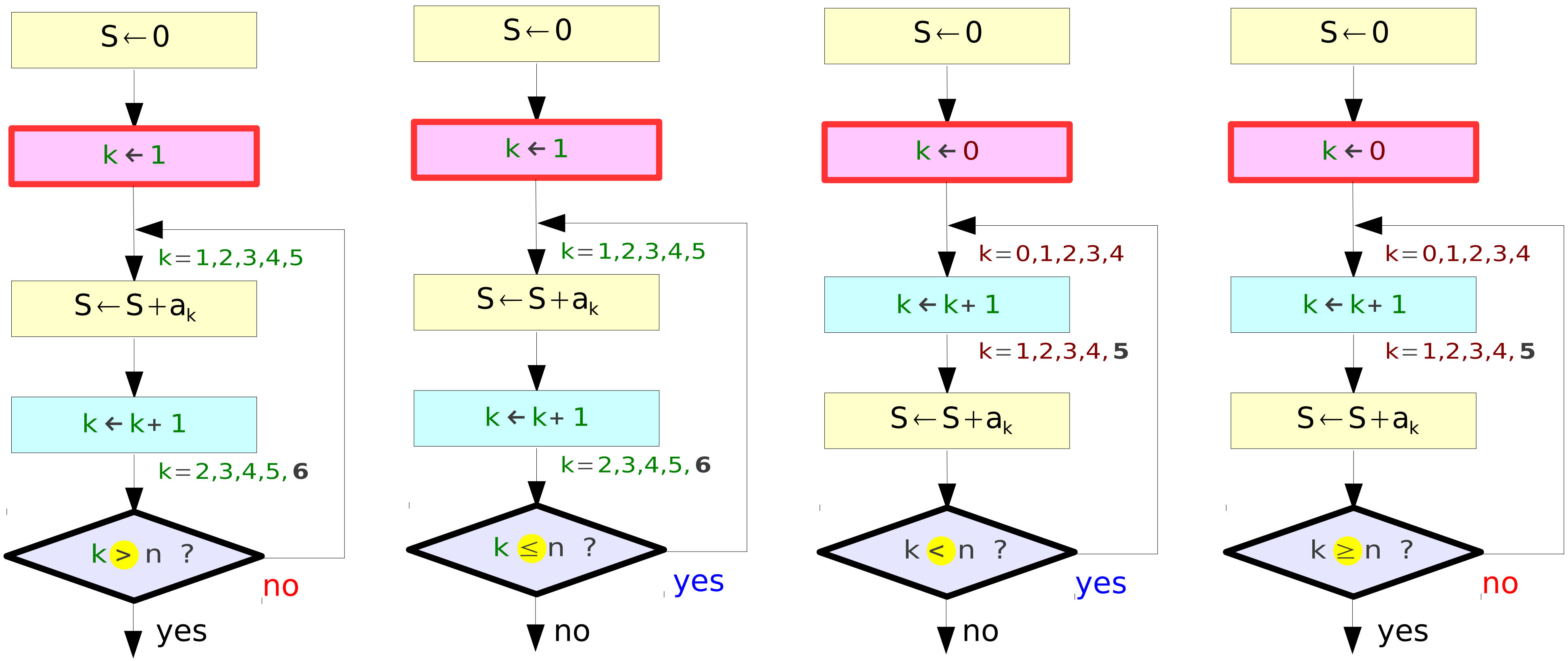


$a_1=2$,
$a_2=4$,
$a_3=6$,
$a_4=8$,
$a_5=10$

Check Condition (Pre-check, Post-check)



Initial Condition (0-start, 1-start)



$S \leftarrow 0$

$k \leftarrow 1$

$k = 1,2,3,4,5$

$S \leftarrow S + a_k$

$k \leftarrow k + 1$

$k = 2,3,4,5,$ **6**

$k > n$?

no

yes

$S \leftarrow 0$

$k \leftarrow 1$

$k = 1,2,3,4,5$

$S \leftarrow S + a_k$

$k \leftarrow k + 1$

$k = 2,3,4,5,$ **6**

$k \leq n$?

yes

no

$S \leftarrow 0$

$k \leftarrow 0$

$k = 0,1,2,3,4$

$k \leftarrow k + 1$

$k = 1,2,3,4,$ **5**

$S \leftarrow S + a_k$

$k < n$?

yes

no

$S \leftarrow 0$

$k \leftarrow 0$

$k = 0,1,2,3,4$

$k \leftarrow k + 1$

$k = 1,2,3,4,$ **5**

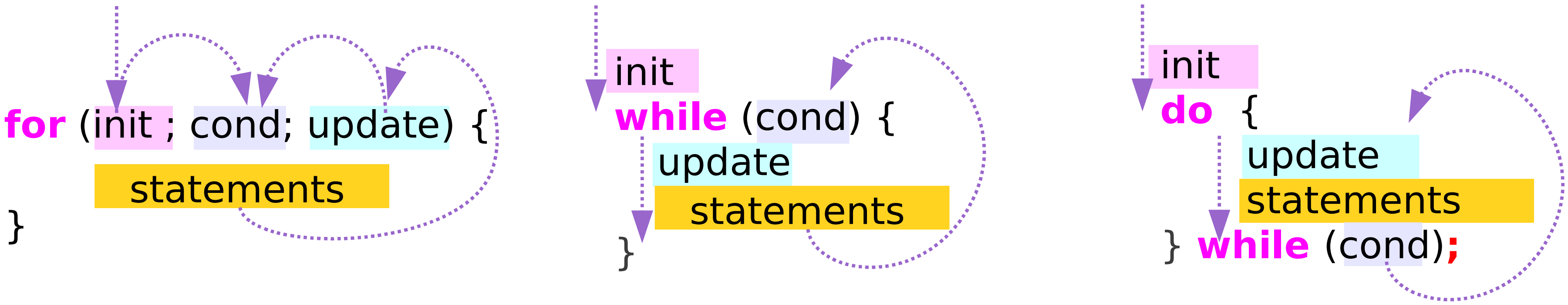
$S \leftarrow S + a_k$

$k \geq n$?

no

yes

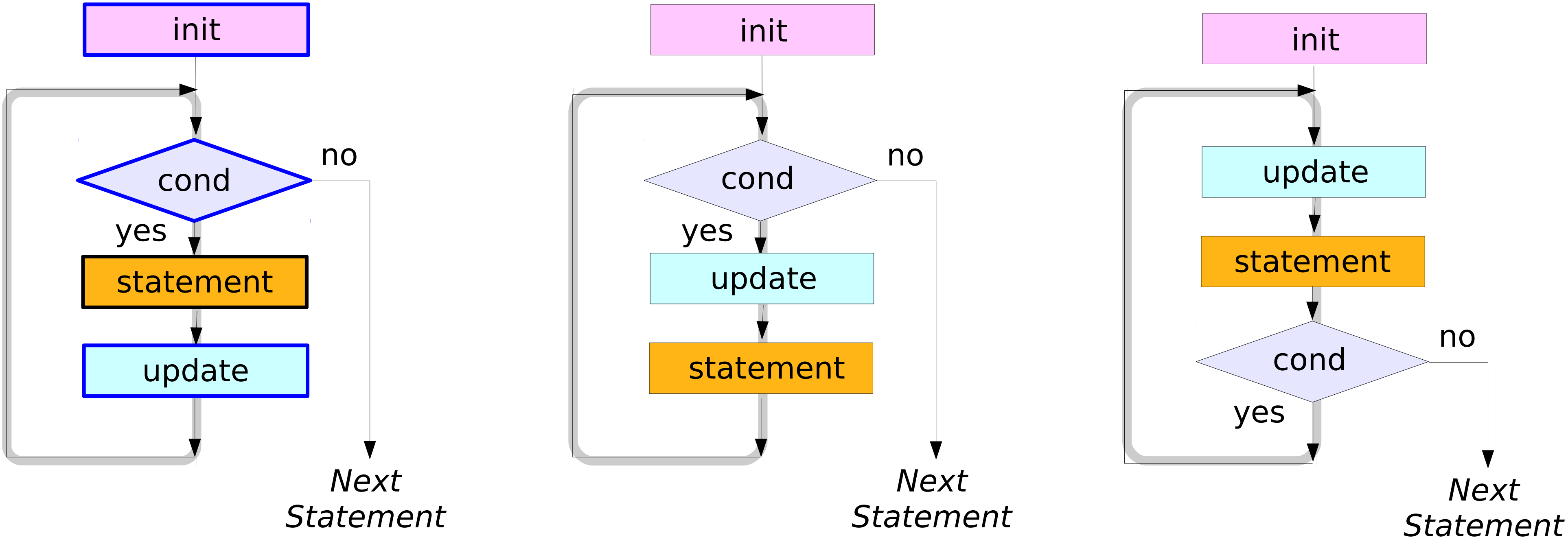
Loop Statements



```
for (init ; cond; update) {
        statements
}
```

```
init
while (cond) {
  update
    statements
}
```

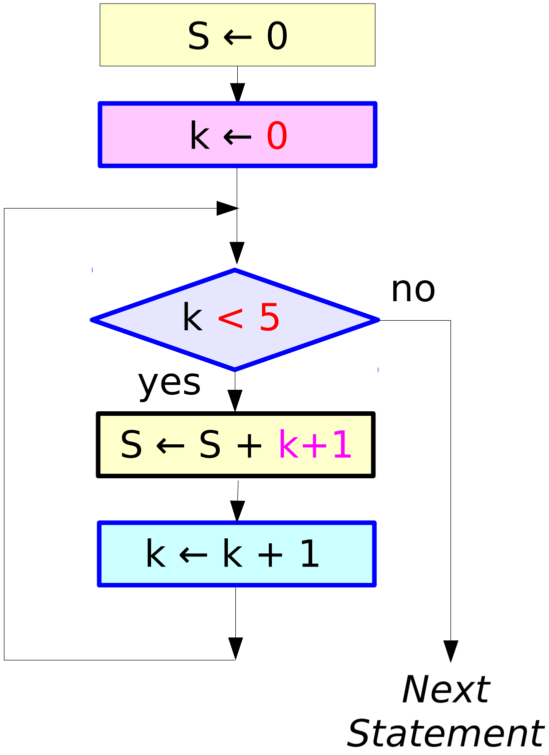
```
init
do  {
  update
  statements
} while (cond);
```



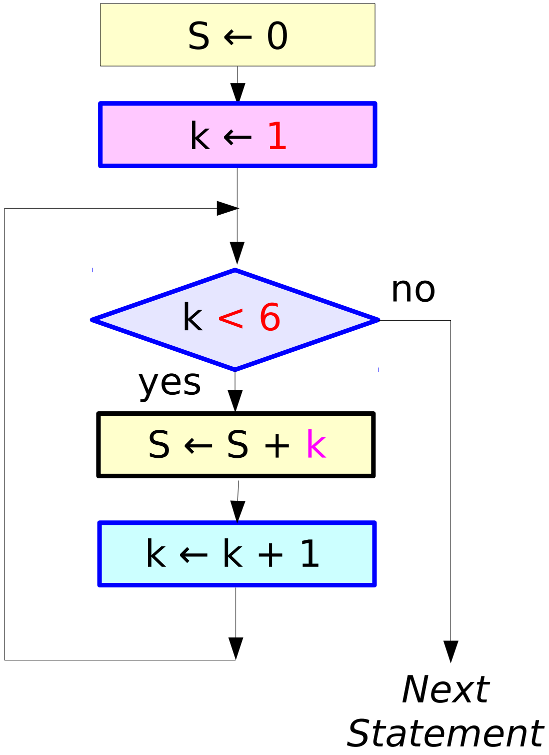
for loop

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

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



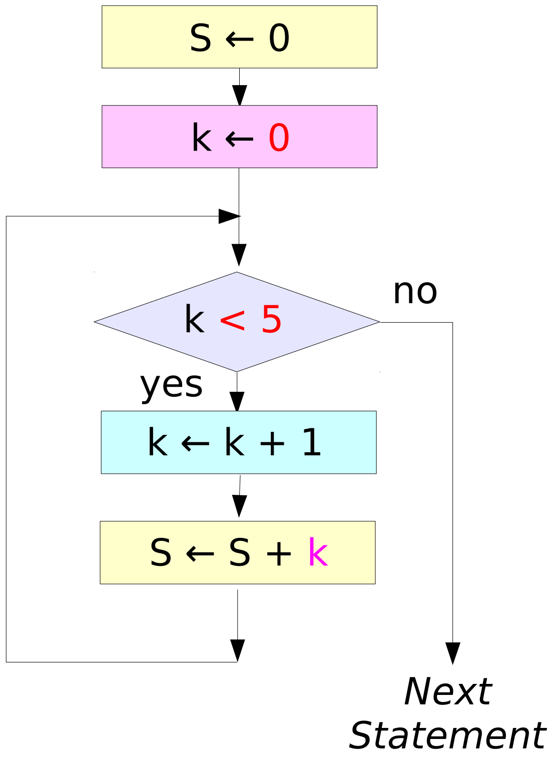
fixed



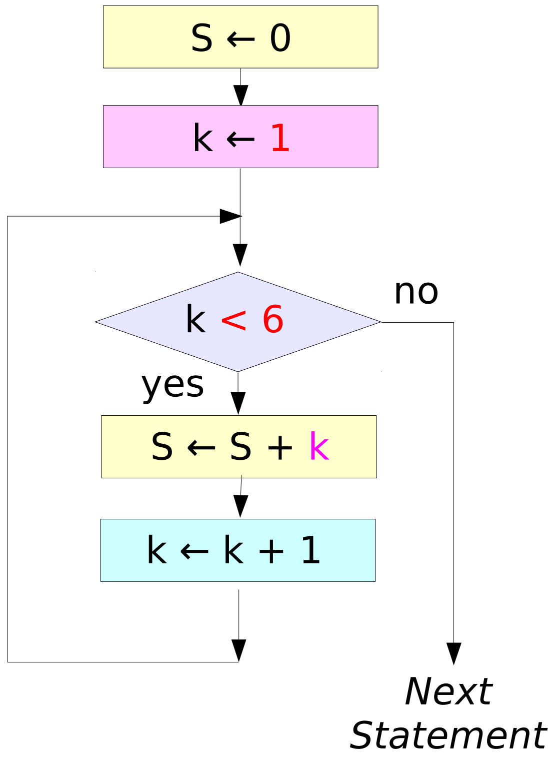
while loop

```
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;
}
```



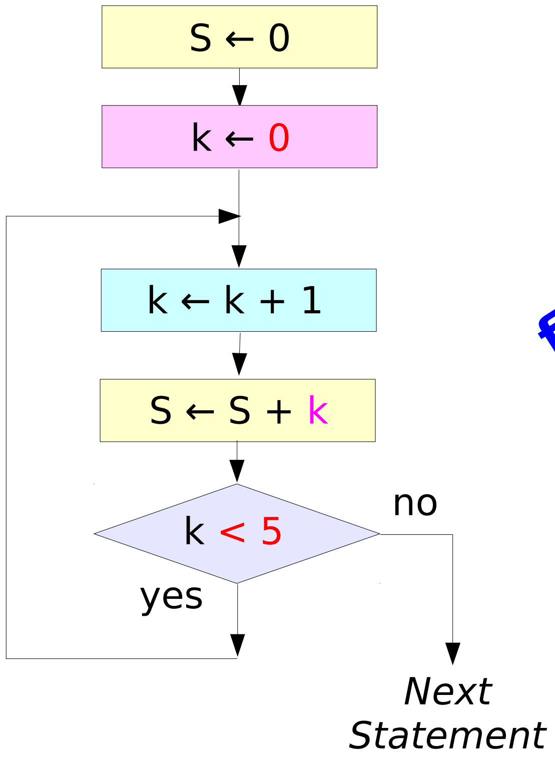
flexible



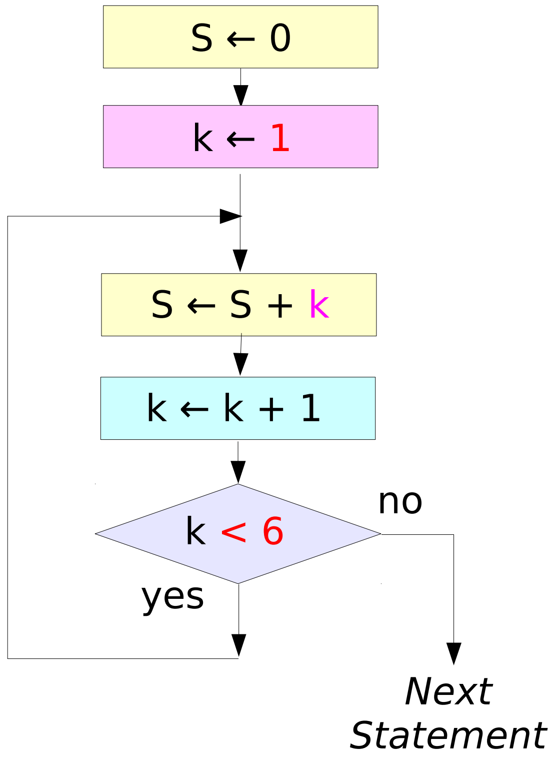
do-while loop

```
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”);
```

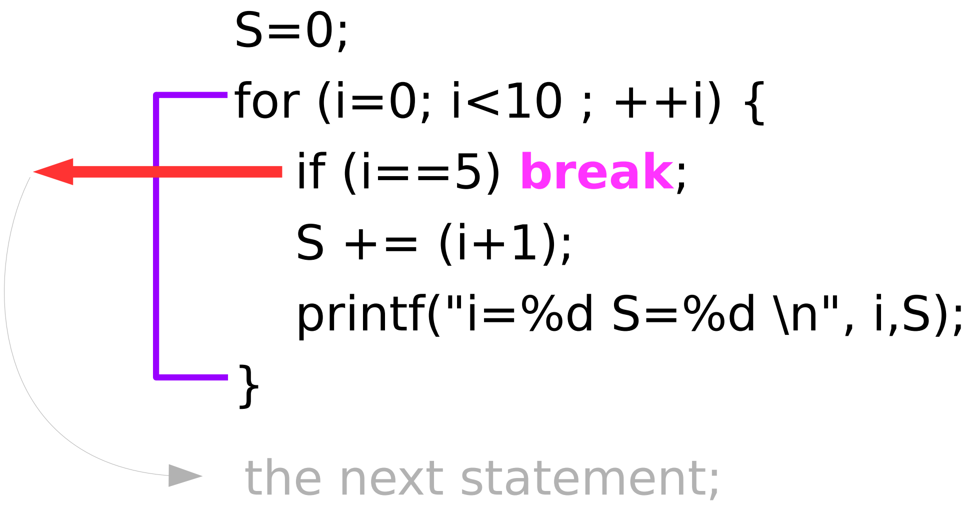
```
for (i=0; i<3; ++i)
printf(“St1 \n”);
for (j=0; j<4; ++j) {
printf(“St2 \n”);
printf(“St3 \n”); }
```

```
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”);
```

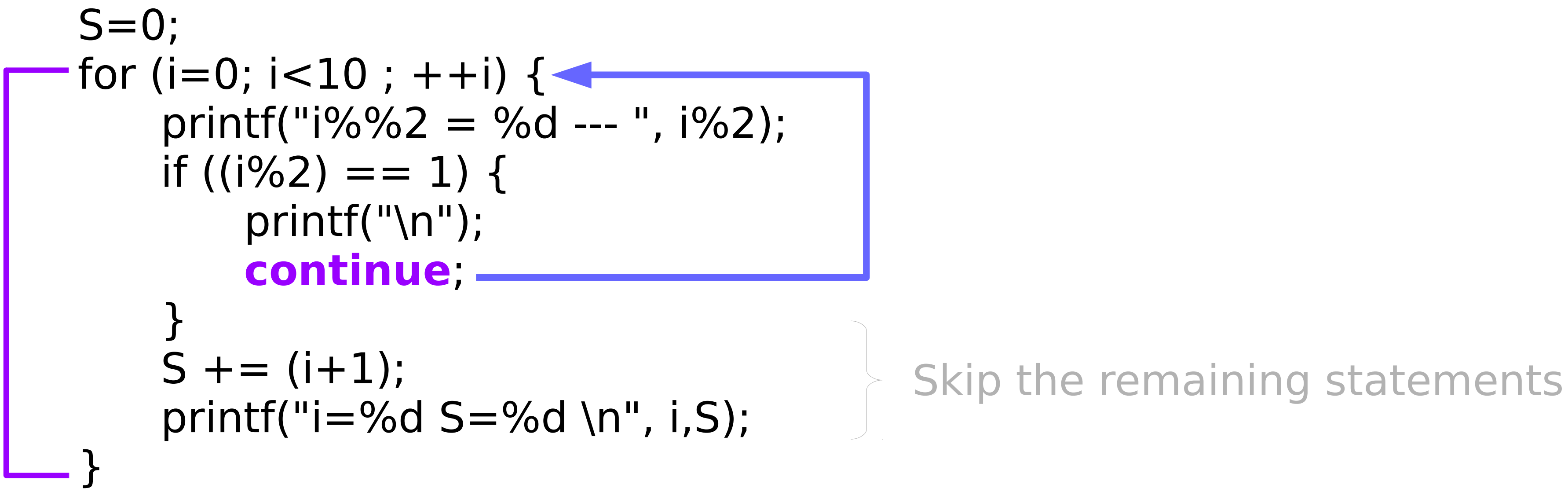
```
for (i=0; i<3; ++i)
        printf(“St1 \n”);
for (j=0; j<4; ++j) {
        printf(“St2 \n”);
        printf(“St3 \n”);
}
```

Break Examples



```
S=0;
for (i=0; i<10 ; ++i) {
    if (i==5) break;
    S += (i+1);
    printf("i=%d S=%d \n", i,S);
}
 the next statement;
```

Continue Examples



```
S=0;
for (i=0; i<10 ; ++i) {
        printf("i%%2 = %d --- ", i%2);
        if ((i%2) == 1) {
                printf("\n");
                continue;
        }
        S += (i+1);
        printf("i=%d S=%d \n", i,S);
}
```

Skip the remaining statements

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