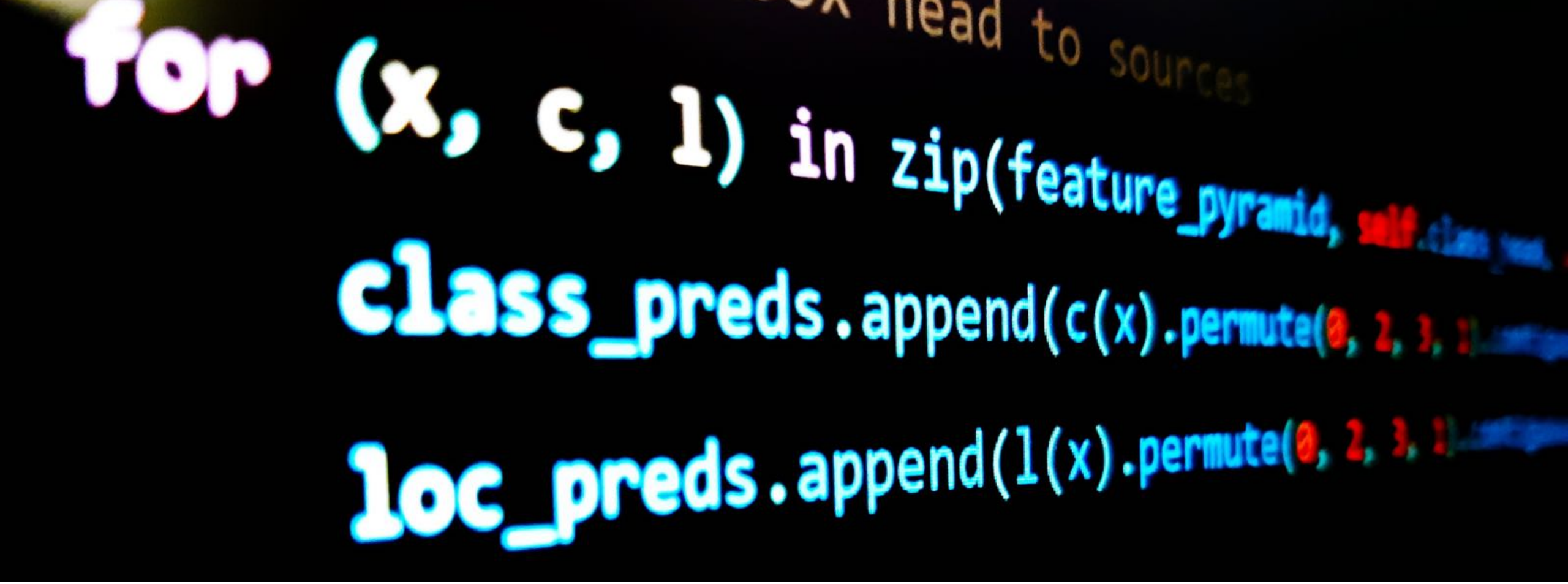


FREQUENCY COUNT

CCDSALG T2 AY 2020-2021

FREQUENCY COUNT

Number of **statements or steps** needed by the algorithm to finish

```
[1] a = 10
[2] b = a * 2
```

```
[1] count: 1
[2] count: 1
```

IF ELSE STATEMENT

```
[1] if <condition>
[2]       <S1>
[3] else
[4]       <S2>
```

$$Total = 1 + max\big(count(< S1 >), count(< S2 >)\big)$$

Note: We compute for the maximum frequency count

IF ELSE STATEMENT

```
[1] if n < 0
[2]       absval = -n
[3] else
[4]       absval = n
```

```
[1] 1
[2] 1          <S1>
[3]
[4] 1          <S2>
```

$$Total = 1 + max(count(< S1 >), count(< S2 >))$$

$$= 1 + max(1, 1)$$

$$= 2$$

IF ELSE STATEMENT

```
[1] if x > 1
[2]       y = 10
[3] else
[4]       y = 20
[5]       z = 30
```

```
[1] 1
[2]     1         <S1>
[3]
[4]     1
[5]     1         <S2>
```

$$Total = 1 + max(count(< S1 >), count(< S2 >))$$
$$= 1 + max(1, 2)$$
$$= 3$$

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] I
[2] I
```

Value of i	Condition	Enter Loop?
0	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] II
[2] II
```

Value of i	Condition	Enter Loop?
1	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] III
[2] III
```

Value of i	Condition	Enter Loop?
2	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] IIII
[2] IIII
```

Value of i	Condition	Enter Loop?
3	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] IIIII
[2] IIIII
```

Value of i	Condition	Enter Loop?
4	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] IIIIII
[2] IIIII
```

Value of i	Condition	Enter Loop?
5	<= 4	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

FOR LOOP

```
[1] for i = 0 to 4
[2]       print(i)
```

```
[1] IIIIII  = 6
[2] IIIII   = 5
```

Value of i	Condition	Enter Loop?
5	<= 4	No

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

```
[1] I
[2] I
```

Value of i	Condition	Enter Loop?
23	<= 25	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

```
[1] II
[2] II
```

Value of i	Condition	Enter Loop?
24	<= 25	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

```
[1] III
[2] III
```

Value of i	Condition	Enter Loop?
25	<= 25	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

```
[1] IIII
[2] III
```

Value of i	Condition	Enter Loop?
26	<= 25	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

FOR LOOP

```
[1] for i = 23 to 25
[2]       print(i)
```

```
[1] IIII    = 4
[2] III     = 3
```

Value of i	Condition	Enter Loop?
26	<= 25	No

Can we derive a formula to get the frequency count of a for loop based on these observations?

FOR LOOP

```
[1] for i = <lb> to <ub>
[2]       <S1>
```

```
[1] <ub> – <lb> + 2
[2] <ub> – <lb> + 1
```

`<lb>` means lower bound, `<ub>` means upper bound

The frequency count for line [1] in detail is:
`<ub> - <lb> + 1 + 1`

The loop condition is true for `<ub> - <lb> + 1` times. The condition becomes false which contributes the other `+ 1`

FOR LOOP

```
[1] for i = 1 to n
[2]       x = x + 1
```

```
[1] n - 1 + 2 = n + 1
[2] n - 1 + 1 = n
```

$$Total = (n + 1) + (n)$$
$$= \mathbf{2n + 1}$$

Note: From hereon, assume that integer n is an input value, and that $n > 0$ unless explicitly specified otherwise.

COMBINATION

```
[1] if x < 1
[2]     y = 10
[3] else if x < 2
[4]     y = 20
[5]     z = 30
[6] else
[7]     for i = 1 to x
[8]         print(i)
```

```
[1] 1
[2]     1
[3] 1
[4]     1
[5]     1
[6]
[7]     x - 1 + 2 = x + 1
[8]     x - 1 + 1 = x
```

Total: $1 + 1 + (x + 1) + x = 2x + 3$

MORE EXAMPLES

```
[1] k = 500
[2] for i = 1 to k - 1
[3]     z = z + 1
```

MORE EXAMPLES

```
[1] k = 500
[2] for i = 1 to k - 1
[3]     z = z + 1
```

```
[1] 1
[2] k - 1 - 1 + 2 = k
[3] k - 1 - 1 + 1 = k - 1
```

MORE EXAMPLES

```
[1] k = 500
[2] for i = 1 to k - 1
[3]     z = z + 1
```

```
[1] 1
[2] k - 1 - 1 + 2 = k
[3] k - 1 - 1 + 1 = k - 1
```

$$Total = 1 + k + (k - 1)$$

$$= \mathbf{2k}$$

MORE EXAMPLES

```
[1] for k = 0 to n
[2]     print(k)
[3]     print(n-k)
```

MORE EXAMPLES

```
[1] for k = 0 to n
[2]     print(k)
[3]     print(n-k)
```

```
[1] n – 0 + 2 = n + 2
[2] n – 0 + 1 = n + 1
[3] n – 0 + 1 = n + 1
```

MORE EXAMPLES

```
[1] for k = 0 to n
[2]     print(k)
[3]     print(n-k)
```

```
[1] n – 0 + 2 = n + 2
[2] n – 0 + 1 = n + 1
[3] n – 0 + 1 = n + 1
```

$$Total = (n+2) + (n+1) + (n+1)$$

$$= \mathbf{3n + 4}$$

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I
[2]
[3]
```

Value of i	Condition	Enter Loop?
1	<= 3	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I
[2] I
[3] I
```

Value of i	Condition	Enter Loop?
1	<= 3	Yes

Value of j	Condition	Enter Loop?
1	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I
[2] II
[3] II
```

Value of i	Condition	Enter Loop?
1	<= 3	Yes

Value of j	Condition	Enter Loop?
2	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I
[2] III-
[3] II-
```

Value of i	Condition	Enter Loop?
1	<= 3	Yes

Value of j	Condition	Enter Loop?
3	<= 2	No

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I
[2] III-
[3] II-
```

Value of i	Condition	Enter Loop?
2	<= 3	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I
[2] III-I
[3] II-I
```

Value of i	Condition	Enter Loop?
2	<= 3	Yes

Value of j	Condition	Enter Loop?
1	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I
[2] III-II
[3] II-II
```

Value of i	Condition	Enter Loop?
2	<= 3	Yes

Value of j	Condition	Enter Loop?
2	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I
[2] III-III-
[3] II-II-
```

Value of i	Condition	Enter Loop?
2	<= 3	Yes

Value of j	Condition	Enter Loop?
3	<= 2	No

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I
[2] III-III-
[3] II-II-
```

Value of i	Condition	Enter Loop?
3	<= 3	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I
[2] III-III-I
[3] II-II-I
```

Value of i	Condition	Enter Loop?
3	<= 3	Yes

Value of j	Condition	Enter Loop?
1	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I
[2] III-III-II
[3] II-II-II
```

Value of i	Condition	Enter Loop?
3	<= 3	Yes

Value of j	Condition	Enter Loop?
2	<= 2	Yes

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I
[2] III-III-III
[3] II-II-II
```

Value of i	Condition	Enter Loop?
3	<= 3	Yes

Value of j	Condition	Enter Loop?
3	<= 2	No

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I-I
[2] III-III-III
[3] II-II-II
```

Value of i	Condition	Enter Loop?
4	<= 3	No

NESTED LOOP

```
[1] for i = 1 to 3
[2]     for j = 1 to 2
[3]         print(i, j)
```

```
[1] I-I-I-I          = 4
[2] III-III-III      = 3 (3)
[3] II-II-II         = 2 (3)
```

Value of i	Condition	Enter Loop?
4	<= 3	No

NESTED LOOP

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         x = x + 1
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1]
[2] (n + 1)
[3] (n)
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1] n + 1
[2] (n + 1) (n)
[3] (n) (n)
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1] n + 1
[2] (n + 1) (n)
[3] (n) (n)
```

$$Total = (n + 1) + (n^2 + n) + (n^2)$$

$$= \mathbf{2n^2 + 2n + 1}$$

NESTED LOOP

```
[1] for i = 2 to n - 1
[2]     for j = 1 to n
[3]         x = x + 1
```

NESTED LOOP

```
[1] for i = 2 to n - 1
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1]
[2] (n + 1)
[3] (n)
```

NESTED LOOP

```
[1] for i = 2 to n - 1
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1] n - 1
[2] (n + 1) (n - 2)
[3] (n) (n - 2)
```

NESTED LOOP

```
[1] for i = 2 to n - 1
[2]     for j = 1 to n
[3]         x = x + 1
```

```
[1] n - 1
[2] (n + 1) (n - 2)
[3] (n) (n - 2)
```

$$Total = (n-1) + (n^2 - n - 2) + (n^2 - 2n)$$

$$= \mathbf{2n^2 - 2n - 3}$$

NESTED LOOP

```
[1] for i = 1 to n
[2]     x = x + 1
[3]     for j = 3 to n + 1
[4]         y = y + 1
[5]         z = z + 1
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     x = x + 1
[3]     for j = 3 to n + 1
[4]         y = y + 1
[5]         z = z + 1
```

```
[1]
[2]
[3] (n)
[4] (n - 1)
[5] (n - 1)
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     x = x + 1
[3]     for j = 3 to n + 1
[4]         y = y + 1
[5]         z = z + 1
```

```
[1] n + 1
[2] n
[3] (n) (n)
[4] (n - 1) (n)
[5] (n - 1) (n)
```

NESTED LOOP

```
[1] for i = 1 to n
[2]     x = x + 1
[3]     for j = 3 to n + 1
[4]         y = y + 1
[5]         z = z + 1
```

```
[1] n + 1
[2] n
[3] (n) (n)
[4] (n - 1) (n)
[5] (n - 1) (n)
```

$$Total = (n+1) + (n) + (n^2) + (n^2 - n) + (n^2 - n)$$

$$= \mathbf{3n^2 + 1}$$

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         for k = 1 to n
[4]             z = z + 1
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         for k = 1 to n
[4]             z = z + 1
```

```
[1]
[2]
[3] (n + 1)
[4] (n)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         for k = 1 to n
[4]             z = z + 1
```

```
[1]
[2] (n + 1)
[3] (n + 1) (n)
[4] (n) (n)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         for k = 1 to n
[4]             z = z + 1
```

```
[1] n + 1
[2] (n + 1) (n)
[3] (n + 1) (n) (n)
[4] (n) (n) (n)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to n
[3]         for k = 1 to n
[4]             z = z + 1
```

```
[1] n + 1
[2] (n + 1) (n)
[3] (n + 1) (n) (n)
[4] (n) (n) (n)
```

$$Total = (n+1) + (n^2 + n) + (n^3 + n^2) + (n^3)$$

$$= \mathbf{2n^3 + 2n^2 + 2n + 1}$$

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     <S1>
[4] while <condition>
```

Different <condition>:

x < <ub>

x ≤ <ub>

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3]
[4]
```

Line 1 will be executed only 1 time.

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3] I
[4]
```

Execute the statement inside the loop without checking the condition.

Value of x
2

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3] I
[4] I
```

Check the loop condition.

Value of x	Condition	Enter Loop?
2	< 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3] II
[4] I
```

Since the condition is true, enter the loop again.

Value of x	Condition	Enter Loop?
3	< 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3] II
[4] II
```

Check the loop condition. Since it is false, exit the loop.

Value of x	Condition	Enter Loop?
3	< 3	No

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] I
[2]
[3] II      = 2
[4] II      = 2
```

Value of x	Condition	Enter Loop?
3	< 3	No

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] 1
[2]
[3] <ub>-<lb> = 2
[4] <ub>-<lb> = 2
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < 3
```

```
[1] 1
[2]
[3] 3 - 1   = 2
[4] 3 - 1   = 2
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x < n
```

```
[1] 1
[2]
[3] n - x = n - 1
[4] n - x = n - 1
```

$$Total = 1 + (n-1) + (n-1)$$

$$= \mathbf{2n - 1}$$

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3]
[4]
```

Line 1 will be executed only 1 time.

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] I
[4]
```

Execute the statement inside the loop without checking the condition.

Value of x
2

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] I
[4] I
```

Check the loop condition.

Value of x	Condition	Enter Loop?
2	≤ 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] II
[4] I
```

Since the condition is true, enter the loop again.

Value of x	Condition	Enter Loop?
3	≤ 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] II
[4] II
```

Check the loop condition.

Value of x	Condition	Enter Loop?
3	≤ 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] III
[4] II
```

Since the condition is true, enter the loop again.

Value of x	Condition	Enter Loop?
4	≤ 3	Yes

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] III
[4] III
```

Check the loop condition. Since it is false, exit the loop.

Value of x	Condition	Enter Loop?
4	≤ 3	No

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] I
[2]
[3] III      =  3
[4] III      =  3
```

Value of x	Condition	Enter Loop?
4	≤ 3	No

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] 1
[2]
[3] <ub>-<lb>+1    = 3
[4] <ub>-<lb>+1    = 3
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ 3
```

```
[1] 1
[2]
[3] 3 – 1 + 1 = 3
[4] 3 – 1 + 1 = 3
```

DO WHILE LOOP

```
[1] x = 1
[2] do
[3]     x = x + 1
[4] while x ≤ n
```

```
[1] 1
[2]
[3] n – x + 1 = n
[4] n – x + 1 = n
```

$$Total = 1 + n + n$$

$$= \mathbf{2n + 1}$$

WHILE LOOP

```
[1] while <condition>
[2]     <S1>
```

Different <condition>:

x < <ub>

x ≤ <ub>

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2]
[3]
```

Line 1 will be executed only 1 time.

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] I
[3] I
```

Value of i	Condition	Enter Loop?
4	< 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] II
[3] II
```

Value of i	Condition	Enter Loop?
5	< 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] III
[3] II
```

Value of i	Condition	Enter Loop?
6	< 6	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] III      = 3
[3] II       = 2
```

Value of i	Condition	Enter Loop?
6	< 6	No

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] <ub>-<lb>+1     = 3
[3] <ub>-<lb>       = 2
```

Value of i	Condition	Enter Loop?
6	< 6	No

WHILE LOOP

```
[1] x = 4
[2] while x < 6
[3]     x = x + 1
```

```
[1] I
[2] 6 – 4 + 1 = 3
[3] 6 – 4     = 2
```

Value of i	Condition	Enter Loop?
6	< 6	No

WHILE LOOP

```
[1] x = 1
[2] while x < n
[3]     x = x + 1
```

```
[1] 1
[2] n – x + 1 = n
[3] n – x = n - 1
```

$$Total = 1 + (n) + (n - 1)$$

$$= \mathbf{2n}$$

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2]
[3]
```

Line 1 will be executed only 1 time.

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] I
[3] I
```

Value of i	Condition	Enter Loop?
4	≤ 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] II
[3] II
```

Value of i	Condition	Enter Loop?
5	≤ 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] III
[3] III
```

Value of i	Condition	Enter Loop?
6	≤ 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] IIII
[3] III
```

Value of i	Condition	Enter Loop?
7	≤ 6	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] IIII     = 4
[3] III      = 3
```

Value of i	Condition	Enter Loop?
7	≤ 6	No

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] <ub>-<lb>+2     = 4
[3] <ub>-<lb>+1     = 3
```

Value of i	Condition	Enter Loop?
7	≤ 6	No

WHILE LOOP

```
[1] x = 4
[2] while x ≤ 6
[3]     x = x + 1
```

```
[1] I
[2] 6 – 4 + 2 = 4
[3] 6 – 4 + 1 = 3
```

Value of i	Condition	Enter Loop?
7	≤ 6	No

WHILE LOOP

```
[1] x = 1
[2] while x ≤ n
[3]     x = x + 1
```

```
[1] 1
[2] n – x + 2 = n + 1
[3] n – x + 1 = n
```

$$Total = 1 + (n + 1) + (n)$$

$$= \mathbf{2n + 2}$$

MORE EXAMPLES

```
[1] x = 1
[2] while x ≤ n
[3]     x = x + 1
```

MORE EXAMPLES

```
[1] x = 1
[2] while x ≤ n
[3]     x = x + 1
```

```
[1] 1
[2] n – 1 + 2 = n + 1
[3] n – 1 + 1 = n
```

MORE EXAMPLES

```
[1] x = 1
[2] while x ≤ n
[3]     x = x + 1
```

```
[1] 1
[2] n - 1 + 2 = n + 1
[3] n - 1 + 1 = n
```

$$Total = 1 + (n + 1) + (n)$$

$$= \mathbf{2n + 2}$$

MORE EXAMPLES

```
[1] x = 1
[2] do
[3]     y = y + 1
[4]     x = x + 1
[5] while x != n - 1
```

MORE EXAMPLES

```
[1] x = 1
[2] do
[3]     y = y + 1
[4]     x = x + 1
[5] while x != n - 1
```

```
[1] 1
[2]
[3] (n - 1) - 1 = n - 2
[4] (n - 1) - 1 = n - 2
[5] (n - 1) - 1 = n - 2
```

MORE EXAMPLES

```
[1] x = 1
[2] do
[3]     y = y + 1
[4]     x = x + 1
[5] while x != n - 1
```

```
[1] 1
[2]
[3] (n - 1) - 1 = n - 2
[4] (n - 1) - 1 = n - 2
[5] (n - 1) - 1 = n - 2
```

$$Total = (1) + (n-2) + (n-2) + (n-2)$$

$$= \mathbf{3n - 5}$$

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to 2n
[3]         x = x + 1
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to 2n
[3]         x = x + 1
```

```
[1]
[2] (2n + 1)
[3] (2n)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to 2n
[3]         x = x + 1
```

```
[1] n - 1 + 2 = n + 1
[2] (2n + 1) (n)
[3] (2n) (n)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     for j = 1 to 2n
[3]         x = x + 1
```

```
[1] n - 1 + 2 = n + 1
[2] (2n + 1) (n)
[3] (2n) (n)
```

$$Total = (n + 1) + (2n^2 + n) + (2n^2)$$

$$= \mathbf{4n^2 + 2n + 1}$$

MORE EXAMPLES

```
[1] for k = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         x = x + 1
```

MORE EXAMPLES

```
[1] for k = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         x = x + 1
```

```
[1]
[2] (n – 4)
[3] (n - 5)
```

MORE EXAMPLES

```
[1] for k = 2 to n + 1
[2]     for j = 3 to n - 3
[3]         x = x + 1
```

```
[1] n + 1 - 2 + 2 = n + 1
[2] (n - 4) (n)
[3] (n - 5) (n)
```

MORE EXAMPLES

```
[1] for k = 2 to n + 1
[2]     for j = 3 to n - 3
[3]         x = x + 1
```

```
[1] n + 1 - 2 + 2 = n + 1
[2] (n - 4) (n)
[3] (n - 5) (n)
```

$$Total = (n+1) + (n^2 - 4n) + (n^2 - 5n)$$

$$= \mathbf{2n^2 - 8n + 1}$$

MORE EXAMPLES

```
[1] for i = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         for k = 4 to n – 4
[4]             x = x + 1
```

MORE EXAMPLES

```
[1] for i = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         for k = 4 to n – 4
[4]             x = x + 1
```

```
[1]
[2]
[3] (n - 6)
[4] (n - 7)
```

MORE EXAMPLES

```
[1] for i = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         for k = 4 to n – 4
[4]             x = x + 1
```

```
[1]
[2] (n - 4)
[3] (n - 6) (n - 5)
[4] (n - 7) (n - 5)
```

MORE EXAMPLES

```
[1] for i = 2 to n + 1
[2]     for j = 3 to n – 3
[3]         for k = 4 to n – 4
[4]             x = x + 1
```

```
[1] n + 1
[2] (n - 4) (n)
[3] (n - 6) (n - 5) (n)
[4] (n - 7) (n - 5) (n)
```

MORE EXAMPLES

```
[1] for i = 2 to n + 1
[2]     for j = 3 to n - 3
[3]         for k = 4 to n - 4
[4]             x = x + 1
```

```
[1] n + 1
[2] (n - 4) (n)
[3] (n - 6) (n - 5) (n)
[4] (n - 7) (n - 5) (n)
```

$$Total = (n+1) + (n^2 - 4n) + (n^3 - 11n^2 + 30n) + (n^3 - 12n^2 + 35n)$$

$$= \mathbf{2n^3 - 22n^2 + 62n + 1}$$

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     j = 2
[3]     while j ≤ n + 3
[4]         print(A[i], A[j-1])
[5]         j = j + 1
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     j = 2
[3]     while j ≤ n + 3
[4]         print(A[i], A[j-1])
[5]         j = j + 1
```

```
[1]
[2]
[3] (n + 3)
[4] (n + 2)
[5] (n + 2)
```

MORE EXAMPLES

```
[1] for i = 1 to n
[2]     j = 2
[3]     while j ≤ n + 3
[4]         print(A[i], A[j-1])
[5]         j = j + 1
```

```
[1] n + 1
[2] n
[3] (n + 3) (n)
[4] (n + 2) (n)
[5] (n + 2) (n)
```

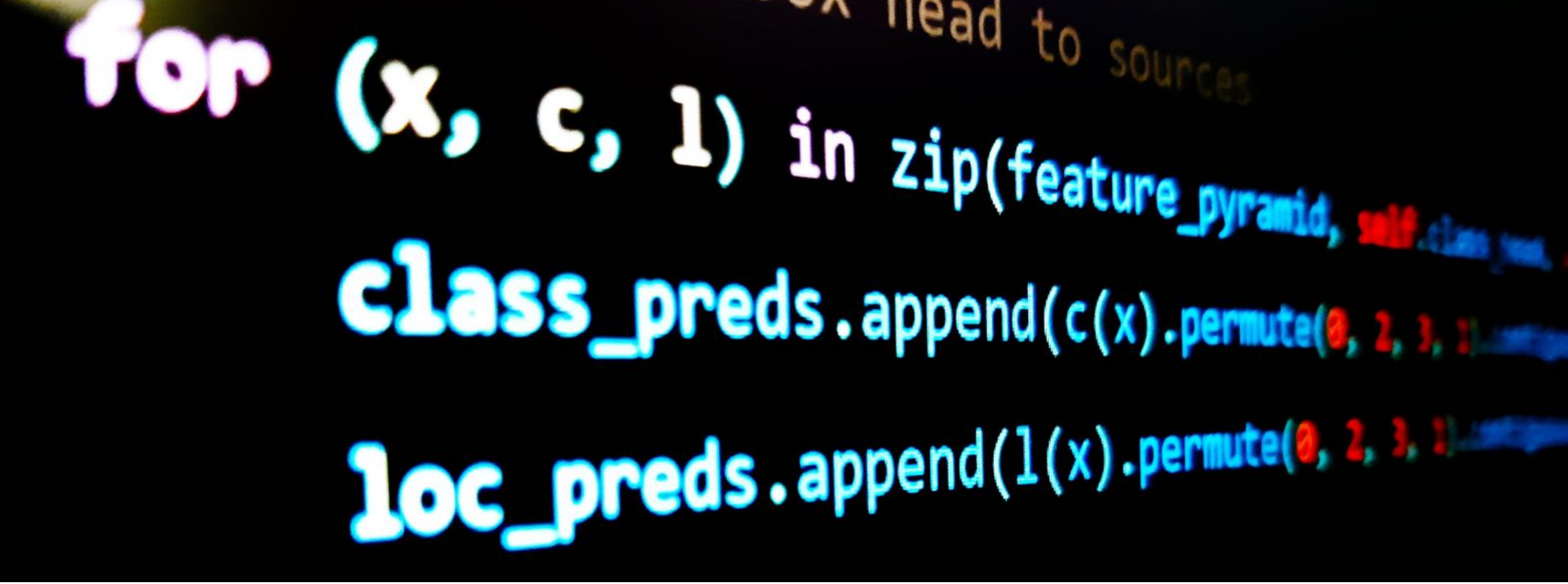
MORE EXAMPLES

```
[1] for i = 1 to n
[2]     j = 2
[3]     while j ≤ n + 3
[4]         print(A[i], A[j-1])
[5]         j = j + 1
```

```
[1] n + 1
[2] n
[3] (n + 3) (n)
[4] (n + 2) (n)
[5] (n + 2) (n)
```

$$Total = (n+1) + (n) + (n^2 + 3n) + (n^2 + 2n) + (n^2 + 2n)$$

$$= \mathbf{3n^2 + 9n + 1}$$



FREQUENCY COUNT

CCDSALG T2 AY 2020-2021