

Nested Loops

Printing a 2-D Figure

How would you print the following diagram?

```
* * * * *  
* * * * *  
* * * * *
```

repeat 3 times

print a row of 5 stars

repeat 5 times

print *

It seems as if a loop within a loop is needed.

H1-2

Nested Loop

```
#define ROWS    3
#define COLS    5
...
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 '*'s */
    ...
    row = row + 1;
}
```

H1-3

Nested Loop

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 '*'s */
    col = 1;
    while ( col <= COLS ) {
        printf( "*" );
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

outer loop: print 3 rows

inner loop: print one row

H1-4

Trace

row:

col:

output:

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-5

Trace

row: 1

col:

output:

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-6

Trace

row: 1

col: 1

output:

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-7

Trace

row: 1

col: 1

output: *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-8

Trace

row: 1

col: 1 2

output: *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-9

Trace

row: 1

col: 1 2

output: * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-10

Trace

row: 1

col: 1 2 3

output: * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-11

Trace

row: 1

col: 1 2 3

output: * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-12

Trace

row: 1

col: 1 2 3 4

output: * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-13

Trace

row: 1

col: 1 2 3 4

output: * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-14

Trace

row: 1

col: 1 2 3 4 5

output: * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-15

Trace

row: 1

col: 1 2 3 4 5

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-16

Trace

row: 1

col: 1 2 3 4 5 6

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-17

Trace

row: 1 2

col: 1 2 3 4 5 6

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-18

Trace

row: 1 2
col: 1 2 3 4 5 6 1

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-19

Trace

row: 1 2
col: 1 2 3 4 5 6 1

output: * * * * *
 *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-20

Trace

row: 1 2
col: 1 2 3 4 5 6 1 2

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-21

Trace

row: 1 2
col: 1 2 3 4 5 6 1 2

output: * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-22

Trace

row: 1 2
col: 1 2 3 4 5 6 1 2 3

output: * * * * *

* *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-23

Trace

row: 1 2
col: 1 2 3 4 5 6 1 2 3

output: * * * * *

* * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-24

Trace

row: 1 2

col: 1 2 3 4 5 6 1 2 3 4

output: * * * * *

 * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-25

Trace

row: 1 2

col: 1 2 3 4 5 6 1 2 3 4

output: * * * * *

 * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-26

Trace

row: 1 2

col: 1 2 3 4 5 6 1 2 3 4 5

output: * * * * *

* * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-27

Trace

row: 1 2

col: 1 2 3 4 5 6 1 2 3 4 5

output: * * * * *

* * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-28

Trace

row: 1 2

col: 1 2 3 4 5 6 1 2 3 4 5 6

output: * * * * *

* * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-29

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6

output: * * * * *

* * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-30

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1

output: * * * * *

* * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-31

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1

output: * * * * *

* * * * *

*

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-32

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2

output: * * * * *

 * * * * *

 *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-33

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2

output: * * * * *

 * * * * *

 * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-34

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3

output: * * * * *

 * * * * *

 * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-35

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3

output: * * * * *

 * * * * *

 * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-36

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4

output: * * * * *

 * * * * *

 * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-37

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4

output: * * * * *

 * * * * *

 * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-38

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5

output: * * * * *

 * * * * *

 * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-39

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5

output: * * * * *

 * * * * *

 * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-40

Trace

row: 1 2 3

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6

output: * * * * *

 * * * * *

 * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-41

Trace

row: 1 2 3 4

col: 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6

output: * * * * *

 * * * * *

 * * * * *

```
row = 1;
while ( row <= ROWS ) {
    /* print a row of 5 *'s */
    col = 1;
    while (col <= COLS) {
        printf("*");
        col = col + 1;
    }
    printf( "\n" );
    row = row + 1;
}
```

H1-42

Print a Multiplication Table

	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9
4	4	8	12

H1-43

Print a Multiplication Table

	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9
4	4	8	12

	1	2	3
1	1 * 1	1 * 2	1 * 3
2	2 * 1	2 * 2	2 * 3
3	3 * 1	3 * 2	3 * 3
4	4 * 1	4 * 2	4 * 3

H1-44

	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9
4	4	8	12

Print Row 2

```
col = 1;
while (col <= 3) {
    printf("%4d", 2 * col);
    col = col + 1;
}
printf("\n");
```

row number

H1-45

Nested Loops

```
row = 1;
while (row <= 4) {
    col = 1;
    while (col <= 3) {
        printf("%4d", row * col);
        col = col + 1;
    }
    printf("\n");
    row = row + 1;
}
```

Print 4 rows

Print one row

H1-46

Loop Trace

row	col	
1	1	print 1
	2	print 2
	3	print 3
		print \n
2	1	print 2
	2	print 4
	3	print 6
		print \n

H1-47

Loop Trace

row	col	
1	1	print 1
	2	print 2
	3	print 3
		print \n
2	1	print 2
	2	print 4
	3	print 6
		print \n

row	col	
3	1	print 3
	2	print 6
	3	print 9
		print \n
4	1	print 4
	2	print 8
	3	print 12
		print \n

H1-48

Counting in *for* Loops

```
/* Print n asterisks */  
for ( count = 1 ; count <= n ; count = count + 1 ) {  
    printf ( "*" ) ;  
}
```

H1-49

Counting in *for* Loops

```
/* Print n asterisks */  
for ( count = 1 ; count <= n ; count = count + 1 ) {  
    printf ( "*" ) ;  
}
```

```
/* Different style of counting */  
for ( count = 0 ; count < n ; count = count + 1 ) {  
    printf ( "*" ) ;  
}
```

H1-50

“3 Rows of 5” as a Nested *for* Loop

```
#define ROWS    3
```

```
#define COLS    5
```

```
...
```

inner
loop:
print
one
row

```
for ( row = 1; row <= ROWS ; row = row + 1 ) {
```

outer
loop:
print 3
rows

```
    for ( col = 1 ; col <= COLS ; col = col + 1 ) {  
        printf( "*" );  
    }
```

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

H1-51

```
1.  /*  
2.   * Illustrates a pair of nested counting loops  
3.   */  
4.  
5.  #include <stdio.h>  
6.  
7.  int  
8.  main(void)  
9.  {  
10.     int i, j;   /* loop control variables */  
11.  
12.     printf("          I      J\n");          /* prints column labels          */  
13.  
14.     for ( i = 1; i < 4; ++i ) {                /* heading of outer for loop          */  
15.         printf("Outer %6d\n", i);  
16.         for ( j = 0; j < i; ++j ) {            /* heading of inner loop          */  
17.             printf("  Inner%9d\n", j);  
18.         } /* end of inner loop */  
19.     } /* end of outer loop */  
20.  
21.     return (0);  
22. }
```

	I	J
Outer	1	
Inner		0
Outer	2	
Inner		0
Inner		1
Outer	3	
Inner		0
Inner		1
Inner		2

```
#define SENTINEL 0
#define NUM_MONTHS 12
```

```
int
main(void)
{
```

```
    int month,          /* number of month being processed */
        mem_sight,     /* one member's sightings for this month */
        sightings;     /* total sightings so far for this month */
```

```
    printf("BALD EAGLE SIGHTINGS\n");
```

```
    for (month = 1;
         month <= NUM_MONTHS;
         ++month) {
        sightings = 0;
        scanf("%d", &mem_sight);
```

```
        while (mem_sight != SENTINEL) {
            if (mem_sight >= 0)
                sightings += mem_sight;
            else
                printf("Warning, negative count %d ignored\n",
                       mem_sight);
            scanf("%d", &mem_sight);
        } /* inner while */
```

```
        printf(" month %2d: %2d\n", month, sightings);
    } /* outer for */
```

```
    return (0);
}
```

```
Input data
2 1 4 3 0
1 2 0
0
5 4 -1 1 0
```

```
/* . . .
```

Results

BALD EAGLE SIGHTINGS

month 1: 10

month 2: 3

month 3: 0

Warning, negative count -1 ignored

month 4: 10

. . .

H1-53

Yet Another 2-D Figure

How would you print the following diagram?

```

*
* *
* * *
* * * *
* * * * *
```

Algorithm:

For every row (**row** = 1, 2, 3, 4, 5)

Print **row** stars

Solution: Another Nested Loop

```
#define ROWS 5
...
int row, col ;
for ( row = 1 ; row <= ROWS ; row = row + 1 ) {
    for ( col = 1 ; col <= row ; col = col + 1 ) {
        printf( "*" ) ;
    }
    printf( "\n" );
}
```

H1-55

Yet One More 2-D Figure

How would you print the following diagram?

```

* * * * *
 * * * *
  * * *
   * *
    *
```

For every row (row = 0, 1, 2, 3, 4)

Print row spaces followed by (5 - row) stars

H1-56

Yet Another Nested Loop

```
#define ROWS 5
```

```
...
```

```
int row, col ;
```

```
for ( row = 0 ; row < ROWS ; row = row + 1 ) {
```

```
    for ( col = 1 ; col <= row ; col = col + 1)
```

```
        printf( " " ) ;
```

```
    for ( col = 1 ; col <= ROWS - row ; col = col + 1)
```

```
        printf( "+" ) ;
```

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

```
}
```

H1-57

Some Loop Pitfalls

```
while ( sum < 10 ) ;  
    sum = sum + 2;
```

H1-58

Some Loop Pitfalls

```
while ( sum < 10 ) ;  
    sum = sum + 2;
```

```
for ( i = 0; i <= 10; i = i + 1);  
    sum = sum + i ;
```

H1-59

Some Loop Pitfalls

```
while ( sum < 10 ) ;  
    sum = sum + 2;
```

```
for ( i = 0; i <= 10; i = i + 1);  
    sum = sum + i ;
```

```
for ( i = 1; i != 10 ; i = i + 2 )  
    sum = sum + i ;
```

H1-60

Use *ints* as Loop Counters

```
int i ;
double x ;
for ( i = 0 ; i < 50 ; i = i + 1 )
{
    x = (double) i / 5.0 ;
    printf("%.18f", x) ;
}
```

H1-61

Exercises: The for Statement: Nested Loop

Write a program to display the following outputs :

```
  *
 ***
*****
*****
```

1



```
  *
 ***
*****
*****
```

2



```
*****
*****
*****
 ***
  *
```

3



```
*****
*****
*****
 ***
  *
```

4



```

#include <stdio.h>
#define COL 7
#define ROWS 4
int
main()
{
    int r,c,i;
    for ( r = 1; r<=ROWS ; r++)
    {
        for (c = 1; c <=4-r; c++)
            printf("#");
        for (i=1; i <= 2* r - 1 ; i++)
            printf("%d",ROWS-c);
        for (c = 1; c <=4-r; c++)
            printf("#");
        printf("\n");
    }

    return(0);
}

```

```

#include <stdio.h>
#define COL 7
#define ROWS 4
int
main()
{
    int r,c,i;
    for ( r = 1; r<=ROWS ; r++)
    {
        for (c = 1; c <=4-r; c++)
            printf(" ");
        for (i=1; i <= 2* r - 1 ; i++)
            printf("%d",ROWS-c);

        printf("\n");
    }

    return(0);
}

```

H1-63