# Tutorial - Week 4

**Objectives:** To practice with

* for, while, do…while repetition statements

1. **What is displayed by this program fragment for an input of 8?**

scanf("%d", &n);

ev = 0;

while (ev < n) {

printf("%3d", ev);

ev = ev + 2;

}

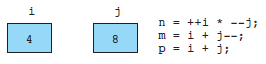
printf("\n");

**Output:**

1. **Write a program fragment that produces this output:**

|  |  |
| --- | --- |
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
|  |  |

1. **Where possible, write equivalents for the following statements using compound assignment operators:**
2. s = s / 5;
3. q = q \* n + 4;
4. z = z - x \* y;
5. t = t + (u % v);
6. **What values are assigned to n, m, and p, given these initial values?**



1. **What errors do you see in the following fragment? Correct the code so it displays all multiples of 4 from 0 through 100.**

for mult4 = 0;

mult4 < 100;

mult4 += 4;

printf("%d\n", mult4);

1. **Show the output displayed by these nested loops:**

for (i = 0; i < 3; ++i) {

printf("Outer %4d\n", i);

for (j = 0; j < 2; ++j) {

printf(" Inner%3d%3d\n", i, j);

}

for (k = 2; k > 0; --k) {

printf(" Inner%3d%3d\n", i, k);

}

}

1. **Write nests of loops that cause the following output to be displayed:**

0

0 1

0 1 2

0 1 2 3

0 1 2 3 4

0 1 2 3 4 5

0 1 2 3 4

0 1 2 3

0 1 2

0 1

0

1. **Rewrite the following code using a do-while statement with no decisions in the loop body:**

sum = 0;

for (odd = 1; odd < n; odd = odd + 2)

sum = sum + odd;

printf("Sum of the positive odd numbers less than %d is %d\n", n, sum);

**In what situations will the rewritten code print an incorrect sum?**

1. **Design an interactive input loop that scans pairs of integers until it reaches a pair in which the first integer evenly divides the second.**

1. **What does the following code segment display? Try each of these inputs: 345, 82, 6. Then, describe the action of the code.**

printf (“\n Enter a positive integer> “);

sacnf(“%d”, &num);

do {

printf("%d ", num % 10);

num /= 10;

} while (num > 0);

printf("\n");

1. **Write a do-while loop that repeatedly prompts for and takes input until a value in the range 0 through 15 inclusive is input. Include code that prevents the loop from executing forever on input of a wrong data type.**