

Algorithms - Loops

1. Write code that lets the user enter a number. The number should be multiplied by 2 and printed until the number exceeds 50. Use a while loop.

2. Write a do-while loop that asks the user to enter two numbers. The numbers should be added and the sum displayed. The user should be asked if he or she wishes to perform the operation again. If so, the loop should repeat; otherwise it should terminate.

3. Write a for loop that displays the following set of numbers:
0, 10, 20, 30, 40, 50 . . . 1000

4. Write a loop that asks the user to enter a number. The loop should iterate 10 times and keep a running total of the numbers entered.

5. Write a nested loop that displays the following output:

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

6. Write a nested loop that displays 10 rows of '#' characters. There should be 15 '#' characters in each row.

7. Rewrite the following code, converting the while loop to a do-while loop:

```
char doAgain = 'y';
int sum = 0;
cout << "This code will increment sum 1 or more times.\n";

while ((doAgain == 'y') || (doAgain == 'Y'))
{
    sum++;
    cout << "Sum has been incremented. Increment it again(y/n)? ";
    cin >> doAgain;
}
cout << "Sum was incremented << sum << " times.\n";
```

8. Rewrite the following code, replacing the do-while loop with a while loop. When you do this you will no longer need an if statement.

```
int number;
cout << "Enter an even number: ";

do
{
    cin >> number;
    if (number % 2 != 0)
        cout << "Number must be even. Reenter number: ";
} while (number % 2 != 0);
```

9. Convert the following while loop to a for loop:

```
int count = 0;

while (count < 50)
{
    cout << "count is << count << endl;
    count++;
}
```

10. Convert the following for loop to a while loop:

```
for (int x = 50; x > 0; x--)
{
    cout << x << "seconds to go.\n";
}
```