Contents

while Loop

This lab serves multiple goals:

- To reinforce your understanding of the syntax of `while` loops,
- To stress the importance of conditions (e.g., the difference between < and <=),
- To study loops whose counter starts at values different from `0`,
- To exhibit that loop counters can be incremented or decremented by any value (not just one),
- To detect and debug infinite loops,
- To design simple algorithms requiring loops, and
- (Optional) To have loops controlled by sentinel values.

Practicing while Loops – Warm-Up

Create a new project, and replace the content of the `Main` method with the following code:

```
int i = 0;
while(i < 100)
{
    Console.WriteLine(i);
    i++;
}
```

1. Execute the code. You should see the numbers 0 to 99 in the console.
2. Replace < with <=, and note that it prints the numbers from 0 to 100, even even though you did not change the numbers.
3. Replace `0` with `100` and `100` with `300`, and note that it prints the numbers from 100 to 300. Observe that the counter can start from and terminate with any number you wish.

4. Modify the code such that it prints all integers between 0 and 100 that are divisible by 3. The solution is given below, but please attempt to do it independently before reading it.

Solution

To implement the above problem, you may use the following:

```
int i = 0;
while(i < 100)
{
    if(i % 3 == 0)
        Console.WriteLine(i);
    i++;
}
```

or

```
int i = 0;
while(i < 100)
{
    Console.WriteLine(i);
    i += 3;
}
```

Which one of the above codes seems more efficient / easier to understand / easier to debug?

Note that you do not have to increment the counter only by one each time. You should update the counter wisely and try to use it more efficiently.

Practicing while Loops – Decrementing Counter

Create a new project and replace the content of the `Main` method with the following code:

```
int n = 100;
while (n > 0)
{
    Console.WriteLine(n);
    n--;
}
```

Execute the code, and explain what you see in the console. Note that the counter is decremented, not incremented.

Practicing while Loops – Mystery Program

Create a new project and replace the content of the `Main` method with the following code:

```
int n;
Console.Write("Enter a natural number greater than 2: ");
n = int.Parse(Console.ReadLine());
int i = 2;
while(n % i != 0 && i < n )
{
    i++;
}
if (i == n)
    Console.WriteLine($"{n} is a ... number");
else
    Console.WriteLine($"{n} is not a ... number");
```

1. What does the code do? Explain the boolean expression of the loop
2. Replace `...` with a meaningful word.

Solution

1. The boolean expression uses a counter `i`, whose original value is 2, and then checks if:
   - the result of the division of `n` by `i` is 0 (stated differently: whether `i` can divide `n`),
   - `i` is less than `n`. In other words, it tries to divide `n` by all the numbers between 2 and `n-1`, and exits if there is a number that divides `n`.
2. This program computes if the number entered by the user is prime! So, we should replace `...` with "prime"!

Practicing while Loops – Summing User-Input

Write a program that asks for an integer value greater than 1 from the user, and computes the result of this series: `1 + 2 + 3 + 4 + ...` up to `n` where `n` represents the number obtained from the user.

Here is an example of execution, where the user input is u n d e r l i n e d, and hitting "enter" is represented by "↵":

```
Please enter an integer greater than 1:
8↵
The sum from 1 to your number is: 36
```

And you can verify for yourself that 1+2+3+4+5+6+7+8 = 36.

Solution

You can look at the code under "Accumulator" at https://princomp.github.io/book.html#vocabulary-1 to get started: essentially, you need to replace the fixed value 10 with the value given by the user.

Infinite Loops

All of the following are examples of infinite loops. Can you spot the "problem"? For each of them, suggest an edit that would make them terminate.

```
int number = 0;
while (number <=5) {
    Console.WriteLine("Hi!");
    Console.WriteLine(number);
}
```

```
int number1 = 0, number = 0;
while (number <=5) {
    Console.WriteLine("Hi!");
    Console.WriteLine(number);
    number1++;
}
```

```
int number = 0;
while (number <=5);
{
    Console.WriteLine("Hi!");
    Console.WriteLine(number);
    number++;
}
```

```
int number = 0;
while (number <=5)
Console.WriteLine("Hi!");
Console.WriteLine(number);
number++;
```

```
int number = 0;
while (number <= 5)
{
    Console.WriteLine("Hi!");
    Console.WriteLine(number);
}
number++;
```

```
int number = 0;
while (number <= 5)
```

```
{
    Console.WriteLine("Hi!");
    number--;
    Console.WriteLine(number);
    number++;
}
```

Pushing Further (Optional)

Here are two advanced **while** loop challenges. Try to think "off-keyboard" for a while before coding your solution, and test it extensively.

Advanced Problem 1

Study the following program:

```
Console.WriteLine("Enter a number to sum, or \"Done\" to
↪  stop and print the total.");
string enter = Console.ReadLine();
int sum = 0;
while (enter != "Done")
{
    sum += int.Parse(enter);
    Console.WriteLine("Enter a number to sum, or \"Done\"
↪  to stop and print the total.");
    enter = Console.ReadLine();
}
Console.WriteLine($"Your total is {sum}.");
```

1. Execute it, and make sure you understand its mechanism.
2. It contains a "sentinel value": can you tell what it is?
3. Write a program by taking inspiration from the previous program. Your program should ask the user to enter integers. After the user indicates they are done (by entering a sentinel value like "Done"), display the smallest value the user entered. If the user did not enter any integers, display "You did not enter anything."

Advanced Problem 2

Write a program that gets a number from the user and finds its biggest divisor less than the number itself.