

Exercise 6 – Looping Constructs

Objective

The major objective is to practice looping within a C program.

Reference Material

This is based on the *Looping Constructs* chapter and language material from both the *Data Types* chapter and the *Expressions* chapter.

This practical session is located in the following directory:

<i>Windows Directory:</i>	c:\qacprogex\looping
<i>Windows Solution directory:</i>	c:\qacprogex\looping\solution
<i>Linux Directory:</i>	/home/user1/qacprg/LOOPING
<i>Linux Solution directory:</i>	/home/user1/qacprg/LOOPING/Solution

Overview

All questions are standalone and can be tackled in any order. (Optional) Question 4 is a thought-provoker and involves nested loops with screen output. (Optional) Question 5 will test your arithmetic skills. Question 3 is used in a later practical exercise on **Functions**.

Practical Outline

1. Open the Visual Studio Solution **numbers.sln**. Write a program that displays the numbers -10 to +10 on the screen, with each number on its own line.
2. Open the Visual Studio Solution **pin.sln**. Write a program that emulates the high-street bank mechanism for checking a PIN, i.e. keep taking input from the keyboard until it is identical to a password number (set by you, the programmer, and held as a `const int`). Implement a mechanism for counting up the number of attempts made before the PIN is correctly entered.
3. Open the Visual Studio Solution **countdown.sln**. Use a loop to ensure that an integer entered by the user at the keyboard is positive. Use a second loop to count down from this integer in steps of 2, displaying each number on the screen until either 1 or 0 is reached, i.e. if the integer 16 (validated) is entered, the output would be:

16 14 12 10 8 6 4 2 0

and if 7 is entered, the output would be:

7 5 3 1

Optional:

4. In this question you will be asked to write a program that draws a large V on the screen. The V should be of user-specified height.

To see what the output should look like, open the **looping\solution\drawv.sln** Visual Studio Solution. Build and run the program to see how the solution program behaves.

Now revert to the "empty" Visual Studio Solution, **looping\drawv.sln**. Write a program that behaves like the solution!

5. Open the Visual Studio Solution **sqr.sln**. Write a program that asks for a whole number (i.e. an `int`) and displays its square using the following property of squares:

i.e. $4 * 4 = \text{sum of the first 4 odd numbers} = 1 + 3 + 5 + 7$

$5 * 5 = \text{sum of the first 5 odd numbers} = 1 + 3 + 5 + 7 + 9$

$n * n = \text{sum of first } n \text{ odd numbers}$