

Repetition and Loops

I. Overview and Objectives

In many programming projects there are steps that need to be repeated to solve a specific problem. Repetition control structures, like the while, for and do/while statements in C are used to accomplish that goal.

The learning objective of this lab is to become familiar with repetition statements in C.

Reading and related topics: Course slides lesson 05. Book chapter 5.

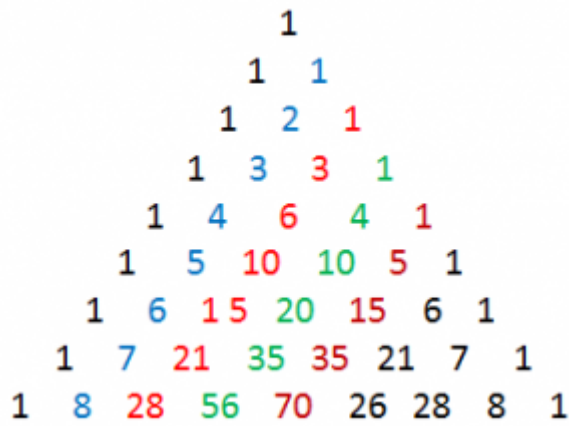
II. Lab Tasks and Submission Guideline

Write complete C programs to solve the following two problems. Save the code you wrote to solve them, together with the result of it in a report. Make sure you include enough comments in your code.

For each problem, copy and paste the source code, copy and paste the execution results of each of the scenarios (best to use *high resolution* screenshots) into your lab report. Save your report in .pdf format and submit it on D2L. You should submit your lab at the end of your lab session or soon after. In all cases it must be submitted before the deadline indicated in the D2L dropbox or it will not be accepted for marking.

Problem 1: The Pascal triangle is a triangular array of the binomial coefficients. To build the Pascal triangle, start with number 1 at the top, then continue placing numbers below it in a triangular pattern, with the second row having two numbers, the third row having 3 numbers, and so on.

Each row starts and ends with number 1, with all the other numbers being the sum of the two numbers directly above them.



Write a C program that uses a loop (or loops) to display the Pascal triangle with 9 rows as shown in the above figure. The numbers must be calculated, not hard coded. The triangle need not to be centered, you can display the triangle this way instead:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1

```

Problem 2: Write an algorithm and C program for the following program definition: Calculate the weekly gross pay for a series of casual workers, using the following information:

- a) For each worker there is a line of data with the following fields, separated by blanks - employee number (integer), number of shifts (integer), base wage rate (real) and one real number per shift, representing the number of hours worked.
- b) Your program should process as many data lines as there are present.
- c) Gross pay is total hours \times base wage rate if the total hours are less than or equal to 15. If the total hours are greater than 15, but less than or equal to 25 there is a 5% premium on base wage rate (for all hours worked). If the total hours are greater than 25 there is a 10% premium.
- d) The gross pay should be rounded to the nearest cent.
- e) Produce a columnar report, with appropriate headings, showing employee number, total hours and gross pay.
- f) Look at the test data provided in [this file](#) to make sure you understand the data format. Determine, by hand calculation, what the correct results should be for this test data. Your program should give correct results for any data file that follows the same format.

Have fun!

