

Assignment #2

Introduction to C Programming – COP 3223

Objectives

1. To give students practice at typing in, compiling and running simple programs.
2. To reinforce knowledge of assignment statements and arithmetic expressions

Introduction: Programmers for a Better Tomorrow

Programmers for a Better Tomorrow is an organization dedicated to helping charities, medical societies, and scholarship organizations manage various tasks so that they can focus on making the world a better place! They have asked you and your classmates to help them develop some new programs to benefit their organizations.

Problem: Knitting for Fun and Non-Profit Part 2 (knit2.c)

One of the programs that Programmers for a Better Tomorrow supports knits hats, sweaters, and blankets for people who may not have enough to keep them warm. These may be sent to other places in the country or overseas.

Each donated blanket is made from 60 knitted squares. These blanket squares are simple and can be made in one day. The BlanketSquares program is becoming very popular and the number of squares made every day is increasing at an exponential rate.

In this program, you will track the number of squares being made in one week using the exponential growth function (shown below).

$$X = a(1 + b)^t$$

a represents the initial population, this is the number of people already making blanket squares on the first of the week. b represents the growth rate, this is the number of new people knitting blanket squares each day. t represents the time interval, which is 7 days for our program.

Your program should calculate X – the number of squares made this week – and print that information for the user.

Then, your program should calculate the number of blankets that can be made from these squares. Print this information and the number of left over squares.

Input Specification

1. The number of people making squares on the first day of the week, a, where a is a positive integer
2. The number of new people making squares each day, b, where b is a positive number value. (This value may be a decimal. If there are three new people every two days, this can be represented by 1.5 people per day)

Output Specification

Output the result using the format below:

X blanket squares will be made this week!

You will be able to make Y blankets and start next week with Z squares.

Output Sample

Below are some sample outputs of running the program. **Note that these samples are NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Run #1

How many people are knitting blanket squares at the beginning of the week?

1

How many new people are knitting blanket squares each day?

.5

17 blanket squares will be made this week!

You will be able to make 0 blankets and start next week with 17 squares.

Sample Run #2

How many people are knitting blanket squares at the beginning of the week?

3

How many new people are knitting blanket squares each day?

2

6561 blanket squares will be made this week!

You will be able to make 109 blankets and start next week with 21 squares.

Deliverables

One source files – *knit2.c* – is to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

- 1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.