CS 420
PROGRAM 1P

Purpose

The purpose of this program is to gain familiarity with C language constructs.

Submission

You need to submit only one C source file called die_roll.c Do not submit any other files.

Scenario

From probability theory the chances of getting any particular number from rolling a 6-sided die is 1/6. If the die is rolled several (hundreds or thousands of) times it would be interesting to see the maximum number of times a particular number occurs consecutively.

Write a program that uses the random number generator in the standard library to simulate the rolling of dice. Run the program for several different rolls of the die and keep track of the longest sequence of each number appearing consecutively i.e. what is the **maximum number of consecutive 1s that appear**, what is **the maximum number of consecutive times 2s appear, etc**. until the total number of attempts are completed. At the end of one run print out the longest sequence (maximum number of consecutive occurrences) for each possible outcome 1 through 6.

Problem

Your program should prompt the user for the number of times you want the dice to be rolled. Use the random number generator for as many times as the dice needs to be rolled (as specified by the user input) and keep track of whether or not the same number was generated in succession, and if so how long was that sequence was. Make sure you only generate numbers 1 through 6 every time. Also use the random number seed at the beginning of the program in such a way that it is as random as possible (using the system time is one way to make sure the seed is quite unique every time) every time the program is run.

Print out the numbers generated as you go along, as well as the totals for the maximum consecutive sequences for each number at the end of the program. Although your program output format does not have to match mine exactly it should be simple enough. Keep the printing of the numbers generated to a maximum of 30 per line. Only integer values are accepted, anything less than 2 is invalid and the user must be prompted again. The program needs to accept only one valid (integer greater than 1) input, and will stop after processing it. You do not need to worry about catching other types of non-integer input.

Sample run of the program is shown below (more of these are in the public folder for 1P):

How many times do you want to roll the dice? 77
Numbers: 3, 1, 3, 3, 5, 4, 6, 1, 6, 5, 1, 6, 3, 3, 5, 6, 1, 5, 3, 3, 1, 1, 2, 6, 4, 4, 3, 4, 6, 6, 5, 6, 4, 5, 6, 2, 2, 6, 1, 5, 4, 1, 4, 4, 3, 3, 1, 2, 5, 2, 4, 3, 2, 3, 2, 6, 6, 2, 1, 4, 6, 3, 1, 3, 1, 1, 3, 3, 4, 3, 1, 5, 1, 3, 6, 2, 3

Consecutive	Count
Ones:	2
Twos:	2
Threes:	2
Fours:	2
Fives:	1
Sixes:	2

Grade Key

Test maximum consecutive values with different input values	**28**
Input validation (9), error inputs followed by valid input (5)	**14**
Clarity of program output	**8**