

LAB 1 Practice Questions

Coverage: Programming in C. Basics – I/O, operators, loops, decision statements.

For this lab you are allowed to use from the C standard library only functions for input and output (e.g. `printf()`, `scanf()`). All programs can be written without using arrays or functions.

1. Write a program to compute the sum of all numbers that are multiples of 4, between 30 and 1000, in three different ways: with a for loop, a while loop and a do-while loop. After each loop print the value on the screen. Write a comment before each loop.
2. Write a program which prompts the user to input a positive number and then it prints it. Your program has to check that the input is positive and if it is not, it has to keep asking for a valid input until one is provided. You may assume that the input is indeed an integer number.
3. Write a program which reads n integers from the standard input (i.e., keyboard), and prints the smallest number. Prompt the user first to input the value of n , which has to be positive. Check if the input is positive and in case it is not, keep asking the user for a correct input. Then prompt the user to input each number.

Hint: In order to find the smallest number, your program should use three variables as follows

- **counter:** A counter to count to n (i.e., to keep track of how many numbers have been input and to determine when all n numbers have been processed)
 - **number:** The current number input to the program
 - **smallest:** The smallest number found so far
4. For this exercise you should be able to write a logical expression (i.e., with logical operators) which checks if some integer x consists of exactly 5 digits. Ex: 30498 and -14004 are 5-digit numbers, while 1098, -1 and 34 are not.

Write a program that reads integers input by the user and outputs those integers of 5 digits. The program should end when 0 is the input. The following is an example of the execution of your program ("↓" is Enter or the Return key)

```
Please input an integer (0 to end): 90807↓
Your input is a five-digit number: 90807
Please input an integer (0 to end): -19253↓
Your input is a five-digit number: -19253
Please input an integer (0 to end): 98↓
Please input an integer (0 to end): 0↓
Thank you for using my software. Bye!
```

5. Write a program that takes a student's average as an input, which is a floating point value, and prints 4 if the average is in the range 90-100, 3 if it is in the range 80-89, 2 if it is in the range 70-79, 1 if it is in the range 60-69 and 0 if the average is between 0 and 59. If the average is not in the range 0-100, the program should print a message to signal an invalid input. You have to use the "switch" statement.
6. Calculate the value of π from the infinite series

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Write a program which reads in a positive integer n (i.e., $n \geq 1$) and calculates the value of π by adding up the first n terms of the above series.

7. A right triangle can have sides that are all integers. The set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find all Pythagorean triples for side1, side2 and the hypotenuse all no larger than 400, with $\text{side1} \leq \text{side2}$. Use a triple-nested for loop that simply tries all possibilities. This is an example of the “brute-force” approach. Your program should print each triple on a separate line, and the total number of triples at the end.
8. A positive integer number is said to be a **perfect number** if its positive factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because $6=1+2+3$. Write a program that prints all perfect numbers smaller than or equal to some integer m ($m > 1$) input by the user.

Note: Assume that x and y are two positive integers. Then x is a **factor** of y if the remainder of the division of y by x is 0. For instance, 5 is a factor of 15, but not of 36.

9. Write a program which reads in a number n and then prints four shapes on the screen, according to the following four patterns

(a)	(b)	(c)	(d)
*****	*	*****	*****
*****	***	*****	***
*****	*****	*****	*
*****	*****	***	***
*****	*****	*	*****

For each shape the number of lines should be n, where n has to be an odd integer between 1 and 25, inclusive. The program should prompt the user to input such an integer, check if this requirement is satisfied, and if it is not satisfied insist until the user inputs a correct integer. You may assume that the input is an integer number.

Note: The number of columns of shape (a) is $2n-1$. Also note that solutions shown are for $n=5$. All solutions should be implemented in one program.

10. Write a program that request the user to input a positive digit and then prints the number of digits in the input. For instance, if the input is 609, your program should print the message:

“Your number has 3 digits”

11. **a)** Write a program that prints backwards a seven-digit positive integer. For example, if the integer is 9806593, the program should print 3956089. You are not allowed to use any function of C standard library other than scanf() and printf(). You are not allowed to use arrays either. For the case when the integer ends with 0, the number printed cannot have leading 0’s (Eg: input 3412400; output 42143). **Hint:** Use the division and remainder operators to separate the number into its individual

digits. **b)** Modify your program so it prints backwards any positive integer, not necessarily a seven-digit one.

12. Write a program that inputs a positive integer containing only 0s and 1s (i.e., a “binary integer”) and print its decimal equivalent. For instance, if the input integer is 10111, the program should print 23 (since $1 + 1*2 + 1*4 + 0*8 + 1*16 = 23$).
13. The factorial of a nonnegative integer n is written $n!$ (pronounced “n factorial”) and is defined as follows: $0! = 1$ and $n! = n*(n-1)*(n-2)*...*1$ for n greater than or equal to 1. For example, $5! = 5*4*3*2*1$, which is 120. a) Write a program that reads a nonnegative integer and computes and prints its factorial. b) Write a program that estimates the value of the mathematical constant e by using the formula $e = 1 + 1/1! + 1/2! + 1/3! + ...$. c) Write a program that computes the value of e^x by using the formula $e^x = 1 + x/1! + x^2/2! + x^3/3! + ...$.
14. Write a program which determines whether a positive integer input by the user, which is different from 1, is prime or not. Note that an integer n , which is not 1, is prime if it has no other factors than 1 and itself.