

Lab Four

ID1063: Introduction to Programming

1. Modify the "sum_of_ten.c" program, so that instead of accepting 10 numbers, you first ask the user for a number n , and then accept n numbers, and print the sum of those n numbers.
2. Write a program to accept a positive integer n , and do each of the following.
 - (a) Print n consecutive * on a line. For example, if $n = 5$, the output is: *****
 - (b) Print all the square numbers in the interval $[1, n]$. For example, if $n = 50$, then the output is 1,4,9,16,25,36,49.
 - (c) Accept a floating point number x and print the first n positive powers of x , i.e. the values $x, x^2, \dots, x^n$, up to 3 decimal places. For example, if $x = 1.1$ and $n = 4$, the values are 1.100, 1.210, 1.331, 1.464.
3. You plan to invest in a scheme where the annual rate of interest is r percent, compounded annually, with $r > 0$. Accept the value r from the user and find the number n of years it takes for the principal amount to at least double.
4. Accept a natural number n , then accept n numbers from the user and print the maximum of the n numbers.

HINT 1: See if you can *keep track* of the maximum after seeing each number in the file example-maximum.pdf

HINT 2: As in the example where we calculated the sum of n numbers without storing them, use a single variable to store the input numbers and use one other variable to keep track of the maximum.

5. Store a positive integer in $[1, 10]$ in a variable x , and ask the user to guess the value. As long as they guess the wrong answer, print "Wrong answer! Try again." and once they print the correct answer, print "You got it!"

6. (a) In the program "divide_by_two.c", divide by 10 instead of by 2 and input a 5-digit number and see what you get.
 (b) Accept a positive integer n and print the digits in decimal representation from right-to-left. For example, if $n = 2026$, the output is 6, 2, 0, 2.
 (c) **Optional:** Print the digits from left-to-right. In the above example, the output would be 2, 0, 2, 6. You DO NOT need arrays for doing this.
7. Change a single line in the program "counting.c" so that it prints the numbers $1, 2, \dots, 10$ in *descending* order.
8. **[Optional]** Accept a real number $x \in (-1, 1)$ and find the value of $\tan^{-1} x$ by adding the first 20 terms of the appropriate Taylor series, which is $\tan^{-1} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$. Compare the values with the values from the in-built function from math.h.

HINT: For each iteration, find the power of x and the denominator separately.

NOTE: To compile using math.h, you must compile as follows: `gcc filename.c -lm -o out`. The option "-lm" tells the compiler to link the math.h library.

9. **[Optional]** Accept a positive integer n and print n consecutive lines of stars, with the i th line having i stars. Example output for $n = 5$:

```
*
**
***
****
*****
```