Problem 1. The purpose of this program is to show that you can already calculate complex functions like the natural logarithm.

Write a program `raphson.c` that uses the Newton-Raphson algorithm to calculate \( \ln(a) \) for a positive `double a`, and compares it to the corresponding value computed by the C math library.

Observe that \( \ln(a) \) is the unique root of the equation \( f(x) = e^x - a = 0 \). You may want to use the fact that \( f'(x) = e^x \), and that \( e^x \) can be approximated by the Taylor series

\[
e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}.
\]

To calculate \( e^x \), consider the first 20 terms in the Taylor series. For Newton-Raphson, continue the iterations until the difference between values in two consecutive iterations is less than 0.001, namely, \( |x_i - x_{i-1}| < 0.001 \). You can model your code after the lecture’s \( \pi \)-calculation example.

Print your computed value of \( \ln(a) \) to 8 decimal digits. Then use the C math library `log` function to compute \( \ln(a) \) again, and print it to the same 8-digit precision. To do that, first write `#include <math.h>` at the top of the program, and then print the value by typing some variation of `printf("%lf", log(a));`. Observe how close the answers are.
Problem 2.
Write a program `closest.c` that asks the user for a 10-element integer array, and prints the two closest elements, namely the two elements whose absolute difference is lowest among all pairs. Note that these elements may not necessarily be consecutive. If multiple pairs have the same lowest difference, print any of them.

```bash
(˜)$ a.out
Array of 10 integers: 13, 19, 65, 24, 35, 46, 57, 78, 99, 21
19 at position 1 and 21 at position 9
(˜)$ a.out
Array of 10 integers: 92, 95, 98, 89, 43, 49, 52, 51, 21, 11
52 at position 6 and 51 at position 7
(˜)$
```
Problem 3.

Write a program `cyclic.c` that first asks the user for an array length ≤ 10, and then requests two integer arrays of the desired length. The program then checks if the second array is a right-cyclic-shifted version of the first one and if so, by how many elements. For example right-cyclic-shifting \([1, 2, 3, 4, 6]\) by 2 places results in \([4, 6, 1, 2, 3]\).