# 1 Overview

This week we shall use dynamically allocated arrays.

# 2 Exercises

- Use the formula below to estimate the value of $e$ by setting $x = 1$. The user inputs the order of the polynomial needed, dynamically allocate an array of type double which stores the the term for each power of $x$ separately. Compare the estimate against the true value. Evaluate the estimates from 1st order to 15th order polynomials. Something strange happens with the higher order polynomials the estimate gets worse, why is that?

  $$f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

  For example, $e$ approximately equals to 2.71828 while $f(1) = 2$, $f(2) = 2.5$, $f(3) = 2.6666$ and so on.

- Basic C Pointers and dynamic memory allocation:

  1. Write a function `allocatearray()` that takes an integer as an argument and returns a pointer to an allocated memory block of that many integers.
  2. Write a function `fillwithones()` that takes a pointer to an array of integers and fills every cell of the array with a one. What other argument do you need to pass to this function?
  3. Write a function `printarray()` that takes a pointer to an array of integers and prints its elements on screen.
  4. Write a function to free the allocated memory.
  5. Call all these functions into a main block. Run with different array sizes.