

Programming Techniques for Scientific Simulations

Exercise 2

Problem 2.1 Static & dynamic arrays (no block assignment)

For this problem you have to write a program reading numbers from the standard input, perform some operations with them, print out the result in reverse order.

The first part of this exercise should be solved using static arrays.

1. Write a program allocating an array of length 10, and let the user provide an undetermined number of values¹ (limited to 10).
2. Normalize the loaded sequence of values, so that the sum is 1.
3. Print out the normalized sequence in reverse order.

Now write a new program, and repeat the previous steps with dynamic arrays, so that the user is no more limited by size of the array.

Problem 2.2 Simpson integration using function pointers as a library
(Block B)

1. Write a C++ function for the Simpson integration which takes the function to integrate, the integration interval and the number of bins as an argument. What are the preconditions and the postconditions? Document your function thoroughly and check the conditions using assertions.
2. Take that function and copy it to a different file. Create a header file that declares the function. Compile and link your program.
3. Create a Makefile that compiles the function for you. Make sure you get the dependencies right. If you do it right, the `make` command should compile only the files that have changed.
4. Create a library `libintegrate.a` that contains your Simpson integration function. Rewrite your Makefile to link against it.

¹The user can close the standard input with Ctrl+D.