2021 Spring   CPSC 240

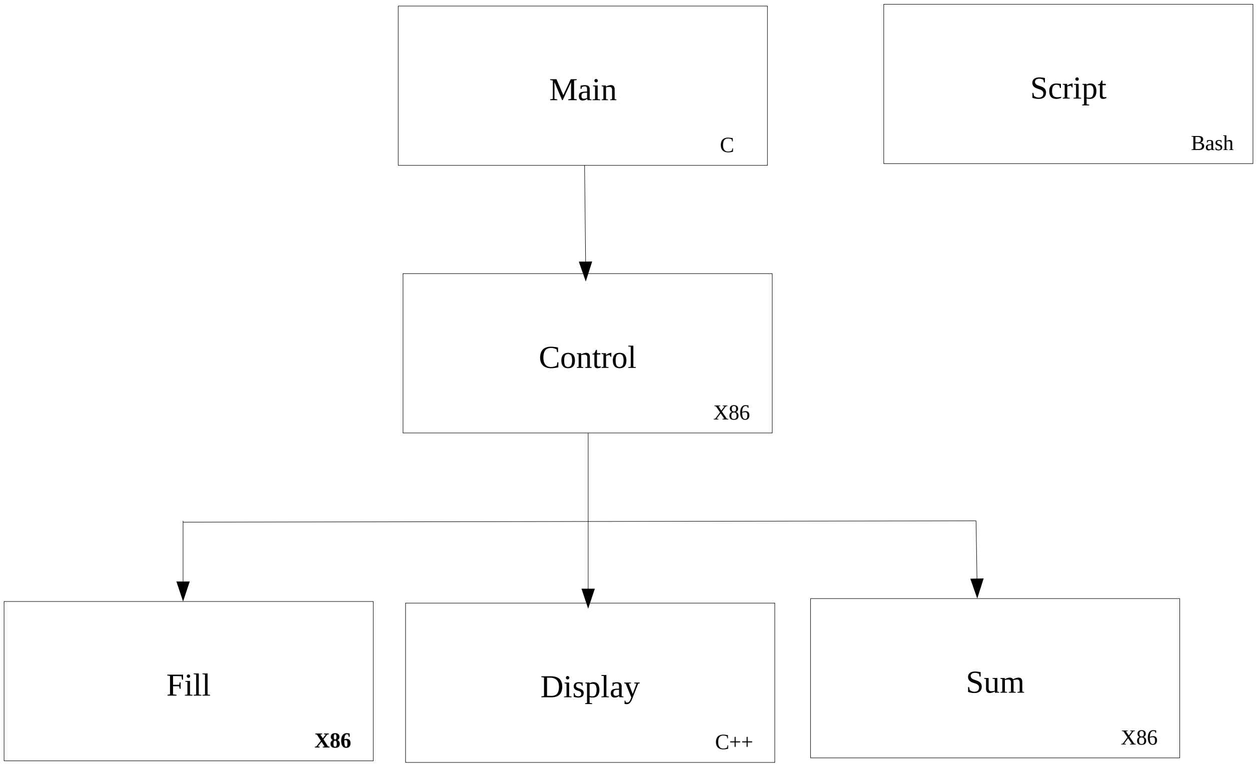
Assignment 3 Sum of an Array

Preface

Create a program in mixed languages (C, C++, and of course x86) that will compute the sum of the float numbers in an array.

Requirements

The following is the calling diagram of the required program.



There are three assembly files, one C++ file, one C file, and one Bash file.

Dialog between User and Program

```
Welcome to High Speed Array Summation by Jose Miracat.
Software Licensed by GNU GPL 3.0
Version 1.0 released on January 28, 2021.
Welcome to HSAS.  The accuracy and reliability of this program is guaranteed by Jose M.
Please enter floating point numbers separated by ws.
When finished press enter followed by cntl+D.
6.5
3.2
-7.4       5.1    2.9
1.0
<cntl+D>
The numbers you entered are these:
6.50000000
3.20000000
-7.40000000
5.10000000
2.90000000
1.00000000
The sum of these values is 11.30000000.
The control module will now return the sum to the caller module.
The main has received this number 11.3000000000 and will keep it.
Thank you for using High Speed Array Software.
For system support contact Jose Miracat at jose@hsas.net
A zero will be returned to the operating system.
```

Color codes
Yellow: Output from driver
Blue: Output from the Control module
Green: Output from the Fill module
Pink: Output from the Display module
The Sum module does not output to standard out device.

Additional requirements

Make the source files meet the standard we refer to as the professional level

There must be six files in the languages indicated in the calling diagram. The script file must execute the entire program correctly out of the box. It will be tested in a Bash system.

In the dialog replace the fake name Jose with your own real name. Replace Jose's fake email address with your real one.

All the float numbers in this assignment are 64-bit float numbers. There is no input validation requirement for this assignment as there was in the previous assignment. [Don't discard your technique of validating inputs. We may need it in the future.]

When outputting a float number always show at least 8 decimal digits on the right side of the decimal point. You decide how many digits to display on the left side of the point. Always show the point. If the float number is 25.000 then do not output it as 25. You may show more than 8 decimal places on the right of the point if desired.