

AERE 361: Lab 4

Sean Hellar

Due 15 FEB 2022

1 Cheat Sheet

Command	Option	What this command/option combination does
float	sum,div,etc	This keeps decimal places in the variables and math functions
gcc	-o	Allows the c code to be compiled and named into an executbale file
=	!	leaving the ! in front of the = creates a “does not equal”
time	programname	shows the total runtime of the program
ctype.h	none	importing this library is espically useful for checking characters
=	+	leaving the + in front of the = creates a sum statement
isalpha	none	checks whether a character is an alphabet or not
nested if loops	none	they are great for checking all conditions are met and deciding what step is next
——	none	signigies or is a parameter statement
int main	()	establishes main loop needed to run any code in c
int	name	intializes the name as a integer type variable, int can be changed for any kind of number

2 Exercise 5 responses

Problem 1 has two reported errors.

- The initializer element is not constant. To fix this I would add the initial `int main (void)` declaration needed for any code to compile.
- There is a missing `;` after `a + b`. To fix this I would add one.
- In order for `printf` to work the code must have `stdio.h`
- Since the sum of the equation exceeds 255 `int sum` must be used

Problem 2 has two reported errors.

- The initializer element is not constant. Same as exercise 1, I would add the initial `int main (void)` declaration needed for any code to compile.
- There is a missing `;` after `a + b`. To fix this I would add one.
- In order for `printf` to work the code must have `stdio.h`
- In order to keep the decimal `float div` must be used. The print statement must now have `%f` to print the decimal.

Problem 3 has two reported errors.

- There is a missing declaration specifier before the string constant and numeric constant. To fix this I would add the initial `int main (void)` declaration needed for any code to compile.
- It adds 8882 instead of the last four zeros. It is not possible to print 20 zeroes. We can use `%Lf` but even that does not have an accuracy of 20 decimal points.

3 Exercise 10

n	Brute Force Time	Gauss Adder Time
1	.042	.036
10	.047	.036
100	.048	.036
1000	.045	.045
10000	.051	.048

Using the time command for both adders shows that the Gauss Adder is faster for smaller numbers, however with larger numbers the Brute Force Adder takes approximately the same amount of time as the Gauss Adder. While the intelligent and brute force method perform similarly, an intelligent program takes up less space than brute force.

Sum force works by starting at zero and adding 1, then adding 2 to that sum, then 3 and so on n amount of times. It is the most simple way to sum from 0 to n .

Gaussian method uses the equation $(n(n+1))/2$. This allows the user to enter any number n and solves the sum in an intelligent way.

4 Sources

Used for Exercise 5. Shows how to keep decimal places in operations.

- https://linuxhint.com/setting_decimal_precision_c_language/

Used for Exercises 7-9. Websites outline the basics of loops, nested loops and check statements.

- <https://www.codesdope.com/c-loop-and-loop/>
- <https://stackoverflow.com/questions/24714287/break-out-of-if-statement/24714523>
- https://www.tutorialspoint.com/c_standard_library/ctype_h.htm