It is known that a child process runs in its own address space, while its parent process runs in its own address space. We will use the Three Sum algorithm (a variation of the Two Sum algorithm, add one more inside loop) to illustrate this:

2			
-4			
3	2 + (-4) + 3 = 1		
5	2 + (-4) + 5 = 3	2 + 3 + 5 = 10	(-4) + 3 + 5 = 4
-2	2 + (-4) + (-2) = -4	2 + 5 + (-2) = 5	(-4) + 5 + (-2) = -1

That is, add the first with the 2nd and the 3rd, then the 1st with the 2nd, then the 4th, then the 1st with the 2nd, then the 5th, and so on. Then increment the middle loop and add the 1st with the 3rd, then the 4th, and so on.

Write a program the spawns a child process from the parent process. In the child process implement the Three Sum algorithm. As we know, the Three Sum algorithm has $O(n^3)$ order and hence takes a long time to complete (especially for larger files). Use the clock( ) to compute the time it takes to run the program. Start the clock at the beginning of the program and end it at the end of the program, then compute the execution time. This may give a surprising result, especially for larger numbers as you wait for the program to execute. Then begin the clock inside (at the beginning) the child process and end it at the end of the child process. Compute the execution time of the child process and note it and the time of the parent process and note that. Try this with the C system wait( ) function (so the parent waits for the child to complete) and without the wait( ) function.

What do you think caused these results? **Explain!** How long did each process take to complete? Explain! You should make a table (and/or graph) of your results and then an objective analysis.

Requirements:
1.) HardCopy of program
2.) Results (comparative results) in the form of screen shots or a table of data
3.) Written explanation of how the program works and an explanation of results, explain the execution times for the parent and the child process. What is going on?