Requirements:



INTRODUCTION:

A scientific calculator is a calculator designed to help us calculate science, engineering, and mathematics problems. It has way more buttons than our standard calculator that just let us do our four basic arithmetic operations of addition, subtraction, multiplication, and division.

All of these extra buttons allow us to work with various kinds of numbers and problems such as these:

* Trigonometry problems
* Scientific numbers that have a multiplication by 10 to certain power.
* Pi problems
* Logarithm problems with base 10 and the natural base
* Probability problems that use the factorial function

USES:

Scientific calculators are used widely in situations that require quick access to certain mathematical functions, especially those that were once looked up in mathematical tables, such as trigonometric functions or logarithms. They are also used for calculations of very large or very small numbers, as in some aspects of [astronomy](https://en.wikipedia.org/wiki/Astronomy), [physics](https://en.wikipedia.org/wiki/Physics), and [chemistry](https://en.wikipedia.org/wiki/Chemistry).

They are very often required for math classes from the junior high school level through college, and are generally either permitted or required on many [standardized tests](https://en.wikipedia.org/wiki/Standardized_test) covering math and science subjects; as a result, many are sold into educational markets to cover this demand, and some high-end models include features making it easier to translate a problem on a textbook page into calculator input, e.g. by providing a method to enter an entire problem in as it is written on the page using simple formatting tools.

We can now use our calculator to solve trigonometry problems involving sine, cosine, tangent, their inverses, and their hyperbolic functions. When working with trigonometric values, we can change the calculations between degrees, radians, and grads. Also, no, we will have access to a button for pi and Euler's constant, e. There are also buttons that allow you to easily calculate exponents to the second, third, or any other power.

When working with scientific numbers, there is an *Exp* button that lets us easily and quickly input scientific numbers.

Engineering problems make use of exponents, logs, and scientific numbers.

Also, all of these types of problems are usually longer expressions that involve several steps to solve by hand. But with the use of a scientific calculator, we can input the whole expression, push the equals button, and the calculator will perform all the calculations we need in the right order. Yes, the scientific calculator computes our problems following the order of operations.

COST AND FEATURES:

4W's AND 1H:

* **Who** – Any student, developer or an enterprise organization those who are looking for an open-source scientific calculator can use.
* **What** – During many large-scale projects the presence of an independent scalable scientific calculator stays missing which might create an issue.
* **When** – Anytime when a free open-source calculator is needed readymade.
* **Where** – This problem happens with almost every developer beginning their first open-source project.
* **How –**  An user needs to run the c file in a c ompiler environment and then finally execute the code after which he has to chose the function he is looking for and feed the arguments to get the result.

**SWOT ANALYSIS:**

Strengths

* Scalable calculator
* Written completely in C language
* Supports multiple functions
* Supports decimal results upto high precissions
* User Friendly

Weakness

* Couldn’t be used without GNU GCC compiler
* It is platform dependent
* Needs an user skilled in c language

Opportunities

* More functions could be added as per demand of user
* Could be used as an additional tool for a more varied calculator
* Improvements for supporting more complex calculations can be added

Threats Analysis

* Hardware Problem
* Better versions might be available

HIGH-LEVEL REQUIREMENT:

|  |  |
| --- | --- |
| **DESCRIPTION** | **CATEGORY** |
| * A user must know to call every function. | Technical |
| * User must know the number of arguments and the arguments itself to be passed during the function call. | Technical |
| * User must be familiar with executing C codes in his/her own machine. | Technical |
| * User must be familiar with reading and writing files in C language if looking for storing data. | Technical |
| * User must have basic idea of double and float type values. | Technical |

LOW-LEVEL REQUIREMENT:

|  |  |
| --- | --- |
| **DESCRIPTION** | **CATEGORY** |
| * User must have knowledge about calculator and how to use it. | Scenario |
| * User must have knowledge about c language. | Scenario |
| * User must have idea of open-source codes. | Scenario |
| * User must be familiar with different types of functions available. | Scenario |
| * User must have idea of Git and GitHub. | Scenario |