Hello to Sub Feature Documentation 😃

What is the Sub Feature?

> Subtraction is one of the basic arithmetic operations. It involves taking one number (the subtrahend) away from another number (the minuend). The result of subtraction is called the difference.

For example: In the equation

```
8 - 3 = 5
```

8 is the minuend, 3 is the subtrahend, and 5 is the difference.

Breakdown of what each line in the driver `SUB_FEATUE.c` do.

```
#include "Sub_Feature.h"
```

- `#include "Sub_Feature.h"`: This line tells the preprocessor to include the contents of the `Sub_Feature.h` header file.This allows the compiler to recognize functions or variables that are defined elsewhere in the project.

```
float subtract(float a, float b)
{
    return a - b;
}
```

- Parameters:
  - The function subtract accepts two arguments, a and b, which are both of type float. This means that the function is designed to work with floating-point numbers (numbers that may have decimals).
- Subtraction:
  - Inside the function, it simply subtracts `b` from `a` using the `-` operator.
- Return Value:
  - The result of the subtraction (a floating-point number) is returned to the caller.

Example Usage:

If you call this function like so:

```
float result = subtract(10.5, 3.2);
```

The function will subtract `3.2` from `10.5` and return `7.3` (i.e., `10.5 - 3.2 = 7.3`).

if you want to read more about the subtractin check this Subtraction.

CMakeLists.txt explanation

> This CMakeLists.txt file sets up a CMake project for building a static library called SubFeature from source files in a directory `SUB_FEATUE.c` Here is a detailed explanation of each line:

```
CMAKE_MINIMUM_REQUIRED(VERSION 3.2)
```

- This line specifies the minimum version of CMake that is required to configure and build the project. In this case, the minimum version is 3.2. This ensures that the features and syntax used in the CMakeLists.txt file are compatible with CMake version 3.2 or later.

```
project(SubFeature C)
```

This line defines the name of the project and the programming language used. In this case:

- `SubFeature` is the name of the project.
- `C` specifies that the project will use the `C programming language`. This tells CMake to look for `*.c` source files and set up the proper flags for C compilation.

```
include_directories(${PROJECT_SOURCE_DIR}/inc)
```

This line adds a directory to the list of directories that the compiler will search for header files (`#include` files). Specifically:

- `${PROJECT_SOURCE_DIR}` is a CMake variable that contains the absolute path to the root directory of the project.
- `/inc` is the relative path to the inc directory within the project, which is assumed to contain the header files for this project.
- `include_directories` tells CMake to add the specified directory `(${PROJECT_SOURCE_DIR}/inc)` to the include path when compiling the C source files. This means that the compiler will be able to find header files located in this directory.

```
file(GLOB Sub_SRC ${PROJECT_SOURCE_DIR}/src/*.c)
```

This line uses the `file(GLOB ...)` command to collect all C source files `(*.c)` in the src folder of the project:

- `file(GLOB ...)` is used to create a list of files that match a given pattern, in this case, all `*.c` files in the src directory.
- `Sub_SRC` is a variable that will hold the list of source files found by the `GLOB` command. These files will be compiled as part of the project.
- `${PROJECT_SOURCE_DIR}/src/*.c` specifies the absolute path to the src directory (using ${PROJECT_SOURCE_DIR}) and then matches all files with the .c extension in that directory.

```

add_library(SubFeature STATIC ${Sub_SRC})
```

This line creates a `static library` from the `C source` files found in the previous step:

- `add_library` is a CMake command that builds a library from the provided source files.
- `SubFeature` is the name of the library being created.
- `STATIC` specifies that the library will be a static library (as opposed to a shared library or dynamic library).
- `${Sub_SRC}` is the list of .c files collected in the previous step. These files will be compiled and linked into the static library SubFeature.

> This configuration generat a static library that can be linked into other projects or applications.😃

So if you want to get the rest of our calculator features you can click here Calculator.