EXPLANATION

I use the C programming language to implement this web service. The task of implementing a calculator is divided into three sub-problems:

1. Implementing a generic stack (declared in *pofix-stack.h*)

2. Implementing calculation of Postfix Notation Expressions using the stack (which is a simpler problem) (declared in *pofix-calc.h*)

3. Implementing calculation of Infix Notation Expressions (by converting infix expressions to postfix, and using the postfix calculator for the actual calculation) (declared in *infix-calc.h*)

A BASE64 to UTF-8 converter has also been implemented. Automated Tests have been implemented to test the entire calculator implementation. The main program 'foo.c' uses socket programming to establish connections with clients sending GET Requests. Invalid GET Requests are rejected, and if there is invalid BASE64 in the query string, or if there are any errors in the calculation, they are indicated in the returned JSON in the HTTP Response. In the case that the calculation was successful, the calculated result is also returned as JSON. The server is able to handle low to moderate load due to forking processes created each time a connection with a client is established.

Elaborations of the code can be found in the comments documented inside the code. The code is hosted at https://github.com/Rafey10/calculus . To run and test the code, simply follow the build and testing instructions found in the README file.

DEPLOYMENT ON AWS

To deploy the web service I simply utilize an AWS EC2 Instance. Following is the documentation of the deployment process:

Initial Setup:

- The process of launching an EC2 Instance is relatively straightforward. I simply choose the latest free-tier Ubuntu server available on AWS and configure the security group to add rules for SSH and HTTP Traffic coming in from 'anywhere' (any IP address).

- I then create and download a key pair (a '.pem' file), storing it in my local file system (which I will later use to SSH into the created Virtual Machine).

- Having launched the instance I am assigned IPv4 public and private IPs and DNSs.

App Deployment:

- To SSH into the machine I simply use the following command from the local directory where my key for the server is stored:

```
$ ssh -i "mykey.pem" ubuntu@ec2-3-127-27-63.eu-central-1.compute.amazonaws.com
```

- I then run the following series of commands to install the required dependencies for the application:

```
# apt-get update
# apt-get install nginx
# snap install cmake
# apt-get install check
# apt-get install build-essential
# apt-get install pkg-config
```

- I then download the git repository where my application code is hosted into the machine:

```
$  git clone  https://github.com/Rafey10/calculus
$  cd calculus && mkdir build && cd build
$  cmake ..
$ make
# systemctl start foo.service
```

- The project is built, tested, and run with the proper NGINX and systemd configuration, as described in the README file in the repository. Starting the application as a configured systemd service allows it to remain running until we decide to stop it (i.e. the web service is served continuously without being dependent on the machine state).

- We can then access the service from anywhere e.g. by pasting the following URL with a BASE64 query string input into a browser like so:

```
http://ec2-3-127-27-63.eu-central-1.compute.amazonaws.com/calculus?query=[input]
```