CS 460 Calculator Protocol

1. This protocol designates how a client may send algebraic operations, later defined, to a server which in turn will send the result of these operations.
2. The packet structure is in the form of:

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Input 1 | Input 2 | Result |

* 1. Operator = 2^3 bits
  2. Input 1 = 2^10 bits
  3. Input 2 = 2^10 bits
  4. Result = 1 bit

1. A client will send a message to the server also known as the client message. The server will respond with a message with the result of the client’s message also known as the server response.
2. The client message may be an algebraic expression, sans variables, or a command to shut down the connection. The server response will always be the result of the client’s message if it’s an algebraic expression or a short message that the client-server connection has been closed if it is the shutdown the connection command.
3. If the client message is not an acceptable algebraic expression or a shutdown command, the “Client Message Not Acceptable” error will trigger.
4. To shut down the connection, the client will send the command:
   1. “quit”

Where “quit” is a string. The server will respond with the message:

* 1. “The connection to server has been closed.”

Where “The connection to server has been closed.” is a string.

1. Only one operation will be accepted by the server. For example, the “sqrt of 5” client message will be accepted, but the “sqrt of 8 \* 2” will not be. If more than one algebraic operation is sent, the server will trigger a “Multiple Operations when Only One Accepted” error.
2. The input range for these equations are all integers from -9999 to 9999. Input outside if this range will trigger a “Input Out of Bounds” error. Input that is not a real integer will trigger a “Input is Not an Integer” error.
3. Acceptable algebraic operations are:
   1. Addition
   2. Subtraction
   3. Multiplication
   4. Division
   5. Raising to a Whole Number Power
   6. Taking Roots
4. The Addition subprotocol will add two inputs in the form of:
   1. “input\_1 + input\_2”

Where input\_1 will be added to input\_2. The “+” sign denotes the addition operation. input\_1 must be separated by a space character before the “+” symbol. input\_2 must be after the “+” symbol and separated from the “+” symbol by a space. Use the example above as a guideline.

1. The Subtraction subprotocol will subtract two inputs in the form of:
   1. “input\_1 - input\_2”

Where input\_1 will be added to input\_2. The “-” sign denotes the subtraction operation. input\_1 must be separated by a space character before the “-” symbol. input\_2 must be after the “-” symbol and separated from the “-” symbol by a space. Use the example above as a guideline.

1. The Multiplication subprotocol will multiply two inputs in the form of:
   1. “input\_1 \* input\_2”

Where input\_1 will be multiplied by input\_2. The “\*” sign denotes the multiplication operation. input\_1 must be separated by a space character before the “\*” symbol. input\_2 must be after the “\*” symbol and separated from the “\*” symbol by a space. Use the example above as a guideline.

1. The Division subprotocol will divide two inputs in the form of:
   1. “input\_1 / input\_2”

Where input\_1 will be divided by input\_2. The “/” sign denotes the division operation. input\_1 must be separated by a space character before the “/” symbol. input\_2 must be after the “/” symbol and separated from the “/” symbol by a space. Use the example above as a guideline.

1. The Raising to a Whole Number Power subprotocol will raise an input to another input in the form of:
   1. “input\_1^input\_2”

Where input\_1 will by raised the power of input\_2. The “^” sign denotes the raising to a whole number power operation. input\_1 is immediately followed by the “^” symbol. input\_2 must be immediately after the “^” symbol. Use the example above as a guide.

1. The Taking Roots subprotocol will take the root of an input in the form of:
   1. “sqrt of input\_1”

Where the root of input\_1 will be taken. The “sqrt of” string denotes the taking roots operation. input\_1 must be separated by a space character after the “sqrt of” string. Use the example above as a guideline.

1. All error detection is left to the server. The client is allowed to send incorrect algebraic operations, e.g. “sqrt of -1”.
2. Any message sent to the server that triggers an error will lead to the following actions:
   1. The server will record which error type was encountered first. Error types are encountered in the order of:
      1. Client Message Not Acceptable Input Out of Bounds
         1. Not within the range of -9999 and 9999.
      2. Input is Not an Integer
         1. Input is either a decimal or an irrational number.
      3. Multiple Operations when Only One Accepted
   2. The server will send the error message back to the client. An error message is a string of the error type.
   3. The server will close the connection.