

+Design plan:

In understanding the problem, I thought about it as taking in a set of characters, and then making sure all those characters are numbers, not worrying about their numerical value until later. I also first checked to make sure that the length of characters was less than two, because that was another criteria that had to be met for the input to be valid. If these conditions were both met then the number was casted into an integer and then checked to see if it was between 0 and 15.

Making sure the other inputs were valid was a bit simpler, as I was just able to check and see if the input wasn't one of the two or three designated inputs and if it wasn't then an error was displayed and the user was prompted again.

+Problems:

One major problem I ran into while testing was not being able to escape the base selection and enter new number prompts. I had them set up as while loops with multiple conditions ORed together like:

```
while(base != "e" || base != "b")
```

and it didn't seem to work. After researching the problem I came across as solution to AND the conditions together instead. I still don't see the logic behind this, but I need to do a little more research on it.

Another problem I encountered was when entering a second or third or fourth etc number, when it was converted it would be outputted odd. I realized that when it was converted and printed using cout, the base for cout was changed, so I simply had to change the base used by cout back to decimal when I was resetting all the other variables at the beginning of the program.

+Testing:

Testing will consist of checking different numbers and other inputs into the initial prompt of the program, making sure that only the correct number 0-15 work. Next will be entering 0-15 to confirm that they are converted into the different bases correctly. Finally entering different inputs into the last two prompts, though these won't take as much to make sure they are correct.

For checking numbers I did just as before and entered the following:

0-15, "hello", 1 5, -1, 05, 15.000000000000000000000001, 2^2, h, 2

For checking the other prompts I entered the following into the first one:

b, o, h, B, O, H, binary, 0, 1, 4, #

and for the second one:

y, n, yes, no, 0, 1

I knew that the outputs of the binary operations were correct, as it was the same code as last time, and I was able to confirm this by simply converting in my head.