Purpose: In this assignment we will be using system service calls to interact with the terminal window and input redirection. Error checking will be performed on the input to ensure proper program execution.

The program will read in a list of decimal values and output their hexadecimal equivalents. You may want to use the macros from assignment #4 (with some modifications) for this assignment.

A template file has been provided with useful constants, variables, some macros, and comments to guide development. You may add additional variables, constants, macros, etc. as desired.

Some input files are included to aid with testing.

Specifications

The program must perform the following tasks (in order):

- Output a header for the program
- Prompt the user for the number of values to convert
  - The number will be in the form of a decimal value
  - Read the value in from the user
    - Ensure this works for both direct console input (typed in by the user) and for redirection (from a file)
  - Ensure the value fits the length requirement (see below)
  - Convert the value to an integer and store it in the appropriate variable
  - Ensure the value fits within the specified minimum and maximum (see template file)
- For each input value required, do the following:
  - Print out a prompt asking for a value
  - Read in the value
  - Convert the value and place it in the array
- Output a label for the converted values
- Output the converted values
  - Output 5 values per line, separated by a space

    - Print an extra line if the number of values is not evenly divisible by 5.

Error Checking

The following errors should be detected. For each error, print out an appropriate message (provided in template) for the user and end the program.

- All Inputs
    - Too Long: The input provided is more than 20 characters (not including linefeeds or nulls).
- Decimal Strings
    - Spaces are allowed before the sign (+/-), between the sign and the start of the numerals (0-9), and after the sequence of numerals.
    - Unexpected Character: Spaces are not allowed between the numerals.
    - Unexpected Character: If not allowed as specified above, the string is invalid.
    - No Digits: The string must contain at least one numeric digit (0-9).
- Array Length
    - The array length must be between the minimum and maximum size specified in the template.

Submission

Once you are satisfied with the program, upload the assembly source code (.asm) file to the assignment page on the class website.

Example Outputs

Working

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
11
Enter decimal value:
1
Enter decimal value:
04
Enter decimal value:
+ 16
Enter decimal value:
- 1
Enter decimal value:
 + 256
Enter decimal value:
- 255
Enter decimal value:
99
Enter decimal value:
487
Enter decimal value:
1000
Enter decimal value:
 3
Enter decimal value:
 - 496
Converted Values
0x00000001 0x00000004 0x00000010 0xFFFFFFFF 0x00000100
0xFFFFFF01 0x00000063 0x000001E7 0x000003E8 0x00000003
0xFFFFFE10
```

Errors

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
0

Error - Program can only convert at least 1 value.
```

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
5000

Error - Program can only convert at most 1,000 values.
```

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
2
Enter decimal value:
+ Five

Error - Unexpected character found in input.
```

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
10000000000000000000000000000000000000000000000000000000000

Error - Input can be at most 20 characters long.
```

```
Number Converter (Decimal to Hexadecimal)

Enter number of values to convert (1-1000):
3
Enter decimal value:
+

Error - Value must contain at least one numeric digit.
```